

# Approximating PageRank locally with sublinear query complexity

Marco Bressan  
TAO, INRIA-CNRS, LRI  
Université Paris-Sud  
Gif-sur-Yvette, France  
marco.bressan@inria.fr

Enoch Peserico  
Dip. Ing. Informazione  
Università di Padova  
Padova, Italy  
enoch@dei.unipd.it

Luca Pretto  
Dip. Ing. Informazione  
Università di Padova  
Padova, Italy  
pretto@dei.unipd.it

## Abstract

The problem of approximating the PageRank score of a node with minimal information about the rest of the graph has attracted considerable attention in the last decade; but its central question, whether it is in general necessary to explore a non-vanishing fraction of the graph, remained open until now (only for specific graphs and/or nodes was a solution known). We present the first algorithm that produces a  $(1 \pm \epsilon)$ -approximation of the score of *any* one node in *any*  $n$ -node graph with probability  $(1 - \epsilon)$  visiting at most  $O(n^{\frac{2}{3}} \sqrt[3]{\log(n)}) = o(n)$  nodes. Our result is essentially tight (we prove that visiting  $\Omega(n^{\frac{2}{3}})$  nodes is in general necessary to solve even an easier “ranking” version of the problem under any “natural” graph exploration model, including all those in the literature) but it can be further improved for some classes of graphs and/or nodes of practical interest – e.g. to  $O(n^{\frac{1}{2}} \gamma^{\frac{1}{2}})$  nodes in graphs with maximum outdegree  $\gamma$ .

# 1 Introduction

This paper presents the first algorithm to approximate the PageRank score of a node visiting only a vanishingly small portion of its graph regardless of the graph’s topology or the specific node under investigation. This introduction briefly reviews PageRank, its local approximation, and the state of the art (respectively in Subsections 1.1, 1.2, and 1.3) before describing the main results of the paper and the organization of its remaining sections (in Subsection 1.4).

## 1.1 PageRank

PageRank [15] has become a classic measure of centrality on graphs and has been named one of the top 10 algorithms in data mining [27]. Originally proposed to rank Web pages according to their importance, it is employed in an ever-growing number of very diverse fields such as spam detection [20], ranking in databases [18], word sense disambiguation [26], credit systems [21], gene ranking [23], and trendsetter identification [24] – to name just a very few.

The PageRank score of a node  $u$  in a graph  $G$  of  $n$  nodes  $u_1, \dots, u_n$  can be easily defined in terms of an abstract *surfer*, who navigates  $G$  starting from an arbitrary node and moving to a new node at each timestep. If the current node is *dangling*, i.e. has no children, the next node is chosen uniformly at random. Otherwise, it is chosen uniformly at random with some positive probability  $(1 - \alpha)$  and chosen uniformly at random among the children of the current node with positive probability  $\alpha$  (the parameter  $\alpha$  is called the *damping factor*). The PageRank score  $P(u_i)$  of  $u_i$  is simply the stationary probability of the surfer being on  $u_i$ . The (row) vector  $\mathbf{P}(\mathbf{u})$  whose  $i^{th}$  component is  $P(u_i)$  is then the (unique) solution of:

$$\mathbf{P}(\mathbf{u}) = \alpha \mathbf{P}(\mathbf{u}) \mathbf{M} + (1 - \alpha) \mathbf{n}^{-1} \quad (1)$$

where  $\mathbf{n}^{-1}$  is the  $n$ -dimensional row vector whose components all equal  $n^{-1}$  and  $\mathbf{M}$  is an  $n$  by  $n$  matrix whose generic element  $M_{i,j}$  equals  $n^{-1}$  if  $u_i$  has no children, and equals  $m^{-1}$  or 0 if  $u_i$  has  $m > 0$  children and  $u_j$  respectively is or is not one of them.

## 1.2 Local approximations

In many cases of practical interest only the PageRank score of a single node or of a very small set of nodes is needed (e.g. when disambiguating between the few possible meanings of a given word [26], when deciding whether an individual email is spam [20], or when ranking the few Web pages relevant to a very focused Web search [16]). However, computing those scores using Equation 1 (see [9] for a survey on the vast literature on the topic) requires knowing and manipulating the entire adjacency matrix of the graph, which for massive and/or constantly evolving graphs such as the Web and many social networks can have a prohibitive cost. It is then natural to ask if one can still obtain a reasonable approximation of the scores of interest (e.g. within a factor  $(1 \pm \epsilon)$ , possibly subject to some small probability of error  $\epsilon$ ) by exploring only a small portion of the graph.

The problem of obtaining such a “local” approximation of PageRank was first posed a decade ago [16], and has attracted considerable attention [19, 2, 8, 7, 14, 11, 10, 13, 22]. Yet, until now the basic question of whether exploring a non-vanishing fraction of a graph is in general necessary to approximate the PageRank score of any one of its nodes remained open. We remark that this problem is fundamentally different from the related one of identifying the highest scoring node(s) of a graph [11, 10], even though there are some common techniques and results: the latter problem requires *sketching* the whole graph, whereas local PageRank computation is one of a growing set of “local graph problems” (such as finding, in the neighbourhood of a *given* node, a small cut set [3] or a well-connected cluster [25]) where the goal is learning some property of a small, given portion of a graph with minimal information about the remainder.

To formalize the problem, one must describe how algorithms can access a graph by defining a set of available *queries*: each query receives an input (possibly empty) and reveals a (typically tiny) portion of the graph in output<sup>1</sup>. In the popular *link server* model [7], a *Neighbourhood*( $u$ ) query returns a list of all parents and a list of all children of the input node  $u$ . In the *jump and crawl* model [12, 11] a *RandomNode*() query (also known as *jump*) returns a random node of the graph, and a *RandomChild*( $u$ ) query (also known as *crawl*) returns instead a child of the input node  $u$  chosen uniformly at random, or the empty set if  $u$  is childless. In the more minimalist *edge probing* model (see e.g. [1]), a query allows one to check whether an arc connects two given nodes.

Queries in the literature share common traits. Since the number of queries needed to solve a problem should measure its “exploration complexity”, a query should only receive as input already-known portions of the graph (given as input of the problem, or returned by previous queries). For the same reason, a query should disclose no information about the global arc structure of the graph. This is captured by the notion of *natural exploration query* (see [13]), that on a graph  $G$  can only return a (possibly empty) connected subgraph of  $G$  which must either depend solely on the nodes of  $G$  but not on its arcs, or depend solely on the set of arcs insisting on a *single* node given as input to the query. Queries of the first type are *global* queries that allow one to discover remote nodes of the graph, while queries of the second type are *local* queries that allow one to assess the graph’s local structure around a given node. *We stress that all exploration queries in the literature are “natural” according to this definition.*

### 1.3 The state of the art

The taxonomy above makes it easier to examine the state of the art. [17, 5, 11, 10] developed a simple but powerful technique to sample nodes from a graph with probability proportional to their PageRank score, using  $O(1)$  *RandomNode*() and *RandomChild*( $\cdot$ ) queries – and showed how to leverage it to obtain a  $(1 \pm \epsilon)$  approximation of the PageRank score  $P(v)$  of a node  $v$  with  $\tilde{O}(\frac{1}{P(v)})$  *RandomNode*() and *RandomChild*( $\cdot$ ) queries (in fact, to obtain such an approximation *simultaneously* for all nodes in a given set with  $\tilde{O}(\frac{1}{P_{\min}})$  queries, where  $P_{\min}$  is the lowest PageRank score of the set). [11, 10] also proved that  $\Omega(\frac{1}{P(v)})$  queries are in general necessary if one is restricted to *RandomNode*() and *RandomChild*( $\cdot$ ) queries; and since at most a vanishing fraction of all nodes in a graph can have PageRank score asymptotically larger than  $\frac{1}{n}$ , breaking the  $\Omega(n)$  query complexity barrier is in general impossible with only *RandomNode*() and *RandomChild*( $\cdot$ ) queries.

[16] introduced the problem of approximating the PageRank score of a given node with few *Neighbourhood*( $\cdot$ ) queries, evaluating some heuristic exploration strategies experimentally with encouraging results. [2] gave an algorithm based on *Neighbourhood*( $\cdot$ ) queries that can be used to approximate the score of a given node; but in the worst case it visits essentially the entire graph. And indeed, [13] proved that using only *Neighbourhood*( $\cdot$ ) queries one in general needs  $\Omega(n)$  queries to obtain a  $(1 \pm \epsilon)$  approximation of the PageRank score of a given node, or even just to *rank* two nodes whose scores differ by a factor greater than  $(1 + \epsilon)$  (i.e. to determine which has the higher score without necessarily computing it).

Some very general results further ruled out the possibility of breaking the linear complexity barrier unless one allows for a small probability of error. [7] proved one needs  $\Omega(n)$  queries for deterministic  $(1 \pm \epsilon)$ -approximation algorithms, and  $\Omega(\sqrt{n})$  for Las Vegas and Monte Carlo ones, in a scenario where algorithms are not limited to *Neighbourhood*( $\cdot$ ) queries (but still under a specific graph exploration model). [14] showed these bounds hold even for the simpler ranking problem, and [13] strengthened them to  $n(1 - o(1))$  queries for any deterministic or Las Vegas

<sup>1</sup>Note that an algorithm might receive in input some “side information” about the graph (e.g. its size or diameter); but since what information is available strongly depends on the specific application, most of the literature (sometimes tacitly) assumes in the default case there is none [2, 6].

algorithm employing only natural queries.

In the light of these negative results, the path leading to sublinear query complexity appears extremely narrow: one must avoid deterministic or Las Vegas algorithms, avoid using only local queries, and avoid using only jump-and-crawl queries.

## 1.4 Our contribution

We present the first algorithm that breaks the linear query complexity barrier of local PageRank approximation for any given node in any given graph, proving:

**Theorem 1.**  $O\left(\frac{1}{\epsilon}\sqrt{\log\left(\frac{1}{\epsilon}\right)} \cdot n^{\frac{2}{3}} \sqrt[3]{\log(n)}\right)$  *RandomNode()* and *Neighbourhood( $\cdot$ )* queries are sufficient to obtain with probability  $(1 - \epsilon)$  a  $(1 \pm \epsilon)$ -approximation of the PageRank score of any node in any  $n$ -node graph.

Our result is essentially tight: any algorithm employing only natural queries that computes with sufficiently high probability a  $(1 \pm \epsilon)$ -approximation of the PageRank score of a node must in general make  $\Omega(n^{\frac{2}{3}})$  queries. In fact, we show that  $\Omega(n^{\frac{2}{3}})$  queries are necessary even for the easier “ranking” version of the problem, where one has to determine with probability  $\frac{1}{2} + \Omega(1)$  which of two nodes  $u$  and  $v$ , whose scores differ by a (multiplicative) factor larger than  $(1 + \eta)$ , has the higher score. The bound on the score approximation then immediately follows, since one could obviously solve the ranking problem if one could obtain  $(1 \pm \epsilon)$ -approximations of the scores of  $u$  and  $v$  with probability  $\frac{1}{2} + \Omega(1)$  for  $\frac{1+\epsilon}{1-\epsilon} < 1 + \eta$ . More formally, we prove:

**Theorem 2.** For any  $\alpha \in (0, 1)$ , any  $\eta > 0$ , and any (Monte Carlo) algorithm employing only natural queries, there exists a graph of arbitrarily large size  $n$  containing two nodes  $u$  and  $v$  with  $P(u) > (1 + \eta)P(v)$  such that the algorithm must perform  $\Omega(n^{\frac{2}{3}})$  queries in expectation to correctly determine that  $P(u) > P(v)$  with probability  $\frac{1}{2} + \Omega(1)$ .

This is the first non-trivial lower bound applying to Monte Carlo algorithms employing any combination of natural queries; and it implies that, to improve substantially on the results of our algorithm, one must restrict the class of graphs or nodes under consideration. For example, for nodes with “large” PageRank score  $\omega(n^{-\frac{2}{3}})$  the technique yielding Theorem 1 can be easily combined with the technique in [5] (a simple implementation of which is given in Appendix A.3) to achieve a query complexity  $O(\frac{1}{P(v)}) = o(n^{\frac{2}{3}})$ . Alternatively, assuming that the maximum outdegree of the graph is  $o(n^{\frac{1}{3}})$  (a reasonable assumption for many graphs modelling social contexts) also allows our algorithm to be simplified into one with  $o(n^{\frac{2}{3}})$  query complexity for any node regardless of its score. This is formalized in the following, more complex but stronger, version of Theorem 1 parametrized on the size and maximum outdegree of the graph and on the score of the node:

**Theorem 1A.**  $O\left(\min\left(\frac{1}{\epsilon^2} \log\left(\frac{1}{\epsilon}\right) \frac{1}{P(v)}, \frac{1}{\epsilon} \sqrt{\log\left(\frac{1}{\epsilon}\right)} \cdot n^{\frac{2}{3}} \sqrt[3]{\log(n)}, \frac{1}{\epsilon} \sqrt{\log\left(\frac{1}{\epsilon}\right)} \cdot n^{\frac{1}{2}} \gamma^{\frac{1}{2}}\right)\right)$  *RandomNode()* and *Neighbourhood( $\cdot$ )* queries are sufficient to obtain with probability  $(1 - \epsilon)$  a  $(1 \pm \epsilon)$ -approximation of the score  $P(v)$  of any node  $v$  in any  $n$ -node graph with maximum outdegree  $\gamma$ .

Like the upper bound of Theorem 1, that of Theorem 1A is asymptotically tight (except possibly for a factor  $O(\sqrt[3]{\log(n)})$  in the case of nodes with score  $o(n^{-\frac{2}{3}})$  in graphs of maximum outdegree  $\omega(n^{\frac{1}{3}})$ ) as a result of the following, more complex but stronger, version of Theorem 2:

**Theorem 2A.** For any  $\alpha \in (0, 1)$ , any  $\eta > 0$ , any functions  $f(n)$ ,  $\gamma(n)$  with  $\frac{1}{n} \leq f(n) \leq 1$  and  $\gamma(n) \geq 1$ , and any (Monte Carlo) algorithm employing only natural queries, there exists a graph of arbitrarily large size  $n$  and maximum outdegree  $O(\gamma(n))$  containing two nodes  $u$  and  $v$  with  $P(u), P(v)$  in  $\Theta(f(n))$  and  $P(u) > (1 + \eta)P(v)$  such that the algorithm must perform  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{\frac{2}{3}}, n^{\frac{1}{2}} \gamma(n)^{\frac{1}{2}}\right)\right)$  queries in expectation to correctly determine that  $P(u) > P(v)$  with probability  $\frac{1}{2} + \Omega(1)$ .

The rest of the paper is organized as follows. Section 2 contains a concise summary of the main ideas used to prove Theorems 2 and 2A. Section 3 contains an in-depth walkthrough of the main ideas and techniques used to prove Theorems 1 and 1A. Section 4 briefly reviews the significance of our results and directions of future work. The appendix contains all the details necessary to complete the proofs – omitted from Sections 2 and 3 to ease the reader’s burden.

## 2 Lower Bounds

This section gives a concise summary of the proof of Theorem 2A; Theorem 2 follows immediately setting  $f(n) \in O(n^{-\frac{2}{3}})$  and  $\gamma(n) \in \Omega(n^{\frac{1}{3}})$ . All details can be found in Appendix A.2.

The proof hinges on the construction of a graph  $G$  of arbitrarily large size  $n$ , formed by two nearly identical subgraphs that one must distinguish in order to decide which of  $u$  and  $v$  has the higher score. The bound on the number of queries is obtained in expectation assuming that the decision must be made on a graph chosen uniformly at random from the family of all graphs isomorphic to  $G$ , and thus applies to at least one specific graph in the family. The two subgraphs (see Figure 1) contain a first level of nodes whose contribution is “damped” by the presence of additional children (allowing one to increase the number of first-level nodes while leaving the scores of  $u$  and  $v$  essentially unchanged), and differ solely for the presence, in one of the two, of a second level of nodes which ensures that the scores of  $u$  and  $v$  are not within a factor  $(1 + \eta)$  of each other.

To distinguish the two subgraphs, one must at least discover some second-level node, and this can only be done by either exploring the first-level nodes via local queries (starting from  $u$  and  $v$  themselves) or invoking global queries that may directly lead to a second-level node. With a careful choice of the number of nodes in each level, and with an adaptation that keeps the construction working for “small”  $f(n)$ , one can always ensure a maximum outdegree  $O(\gamma(n))$ , scores of  $u$  and  $v$  in  $\Theta(f(n))$  but not within a factor  $(1 + \eta)$  of each other, and an expected  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{\frac{2}{3}}, n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}\right)\right)$  queries to find any second-level node and thus to determine which of  $u$  and  $v$  has the higher score with probability  $\frac{1}{2} + \Omega(1)$  – proving the theorem.

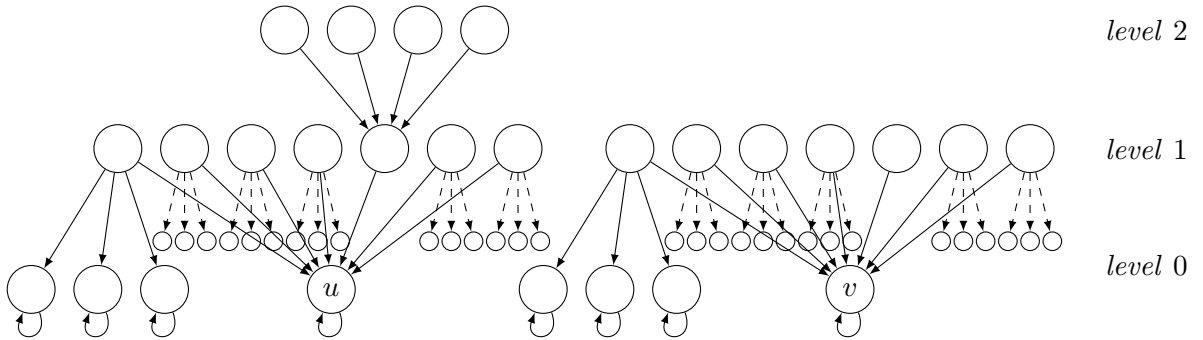


Figure 1: The structure of the generic graph used in the proof of Theorem 2A. To decide which of  $u$  and  $v$  has the higher score, one must distinguish the two nearly-identical subgraphs.

## 3 Upper Bounds

This section provides a detailed walkthrough of the main ideas and techniques behind the proof of Theorem 1A, of which Theorem 1 is a corollary (the next three paragraphs provide a very high-level sketch). Due to the complexity of the proof, we have moved some of the more technical details to the Appendix to ease the burden on the reader.

The starting point of our proof is the simple routine originally delineated in [17, 5] that allows one to sample a node with probability proportional to its PageRank score and thus estimate the

PageRank score of a node or group of nodes by repeated trials. One crucial observation we make is that the routine can also be used to estimate linear combinations of PageRank scores with a variance that, *depending on the coefficients*, can be significantly lower than that of estimating the individual scores. We can leverage this observation by showing how the PageRank score of a generic node can be reformulated as a function of various (in general, uncountably many) linear combinations of the PageRank scores of its ancestors – and of some additional quantities, such as the size of the graph and the aggregate score of all dangling nodes, that can be estimated efficiently through random sampling.

The basic idea of our algorithm is then to explore through *Neighbourhood*( $\cdot$ ) queries the ancestor set of the node of interest, trying to obtain a subset of ancestors yielding a linear combination with “good” coefficients (or, more precisely, a representative sample of such a subset) without using too many queries. Roughly speaking, the nodes in this subset should be biased towards high PageRank scores and high “conductance” to the node of interest, and the coefficients should be chosen in such a way that the contributions of the various nodes are sufficiently balanced.

The obvious difficulty lies in the fact that we cannot in general explore the entire ancestor set of  $u$ , but whether a given portion is worth exploring and would yield “good” coefficients depends in part on information, such as the outdegrees of its nodes, that we obtain only after exploring it. However, we show how this information can be sufficiently approximated by some more global sampling; on the other hand, *knowing* this information beforehand, e.g. because we know that the graph has a low maximum outdegree, can simplify the algorithm and reduce the number of necessary queries (hence the tighter bounds in Theorem 1A compared to Theorem 1).

Let us now delve into the proof.

**Sampling nodes with probability proportional to the PageRank score.** One cornerstone of our algorithm is a simple node-sampling routine (originally delineated in [17, 5]) which emulates PageRank’s random walk using *RandomNode*() and *RandomChild*( $\cdot$ ) queries; note that we can “simulate” an invocation of *RandomChild*(*node*) with a single invocation of *Neighbourhood*(*node*). The routine returns node  $v$  with probability exactly  $P(v)$  and uses  $O(\frac{m}{1-\alpha})$  queries with high probability when called  $m$  times, as stated in the following Lemma 1. A proof of the lemma (including the routine definition) can be found in Appendix A.3.

**Lemma 1.** *There exists a routine *SampleNode*() which returns node  $v$  with probability equal to  $P(v)$  using only *RandomNode*() and *RandomChild*( $\cdot$ ) queries. One call to *SampleNode*() performs less than  $\frac{2}{1-\alpha}$  queries in expectation, and the probability that  $m$  calls to *SampleNode*() perform more than  $\frac{2(1+\Delta)m}{1-\alpha}$  queries is less than  $e^{-\frac{m}{2} \cdot \frac{\Delta^2}{1+\Delta}}$ .*

Combining Lemma 1 with the Chernoff-Hoeffding bounds in Appendix A.1, one immediately obtains the following:

**Corollary 1.** *For any node  $v$  with PageRank score  $P(v)$ , and any arbitrarily small  $\epsilon > 0$ , there exists a number of queries  $q(v) = O(\frac{1}{P(v)})$  such that with  $q(v)$  queries one can estimate  $P(v)$  within a factor  $(1 \pm \epsilon)$  with probability  $1 - \epsilon$ .*

**How large is the graph?** Estimating a PageRank score through *SampleNode*() does *not* require knowledge of the number  $n$  of nodes in the graph. However, this knowledge is required by the full algorithm.  $n$  can be estimated through repeated use of *RandomNode*() with  $\Theta(\sqrt{n})$  queries by a birthday argument, and again exploiting the probability bounds from Appendix A.1, one obtains that for any  $\epsilon > 0$ , the probability of an estimate of  $n$  off by more than a factor  $(1 \pm \epsilon)$  can be made smaller than  $\epsilon$  with  $O(\sqrt{n})$  queries. This allows us to combine *SampleNode*() with other algorithms requiring knowledge of  $n$  and using a number  $f(\cdot)$  of queries that may be larger than that required by *SampleNode*() for some values of  $n$  and  $P(v)$  but smaller for

others, obtaining an overall asymptotic cost of  $\min\left(\frac{1}{P(v)}, \max(\sqrt{n}, f(\cdot))\right)$ , as follows. We invoke *SampleNode()* until either we obtain a “good” estimate of  $P(v)$  through it (we know this is the case with probability arbitrarily close to 1 once *SampleNode()* has returned  $v$  a sufficiently large number  $s \in \Theta(1)$  times), or we obtain a “good” estimate of  $n$  exploiting the invocation of *RandomNode()* made by each invocation of *SampleNode()*. In the latter case, we know  $P(v) = O(\frac{1}{\sqrt{n}})$ , and we can run “in parallel” *SampleNode()* and the alternative algorithm, until either one provides a sufficiently good estimate of  $P(v)$ . We shall now see one such alternative algorithm that estimates  $P(v)$  with  $O(n^{\frac{2}{3}} \sqrt[3]{\log(n)})$  queries; we can then combine it with *SampleNode()* to obtain a hybrid algorithm providing a “good” estimate of  $P(v)$  with  $O\left(\min\left(\frac{1}{P(v)}, n^{\frac{2}{3}} \sqrt[3]{\log(n)}\right)\right)$  queries, thus proving not only Theorem 1 but also Theorem 1A.

**PageRank conductance.** A second cornerstone of our algorithm, crucial for exploring the neighbourhood of the node of interest and re-writing its PageRank score as an “appropriate” linear combination of PageRank scores of its ancestors, is the notion of (PageRank) *conductance* from a node  $u$  to a node  $v$  within a subgraph  $G'$  of a graph  $G$  – informally, the probability that a random walk will carry one from  $u$  to  $v$  without ever leaving  $G'$  or taking a random jump. More formally, given  $G'$  and two nodes  $u$  and  $w$  let us write  $u \xrightarrow{G'} w$  if  $u$  is a parent of  $w$  in  $G'$ , i.e. if there is a link in  $G'$  from  $u$  to  $w$ . Then:

**Definition 1.** Given a node  $v_0$  and a node  $v_f$  in a graph  $G$ , a path  $\pi_{v_0, v_f}^{G' \subseteq G}$  in  $G' \subseteq G$  from  $v_0$  to  $v_f$  is a sequence of arcs of  $G'$  such that either  $\pi_{v_0, v_f}^{G' \subseteq G} = \emptyset$  and  $v_0 = v_f$ , or  $\pi_{v_0, v_f}^{G' \subseteq G}$  is the concatenation of a path in  $G'$  from  $v_0$  to some node  $v_i \xrightarrow{G'} v_f$  with the arc  $(v_i, v_f)$ .

We denote by  $|\pi_{v_0, v_f}^{G' \subseteq G}|$  the *length* of  $\pi_{v_0, v_f}^{G' \subseteq G}$ , i.e. the number of arcs in it; and we abuse the notation slightly and write  $v \in \pi_{v_0, v_f}^{G' \subseteq G}$  if a generic node  $v$  has an (incoming or outgoing) arc  $e \in \pi_{v_0, v_f}^{G' \subseteq G}$ . Then, denoting by  $\text{outdegree}_G(u)$  the outdegree of  $u$  in  $G$ , we can define conductance as follows:

**Definition 2.** The *conductance* of a path  $\pi_{v_0, v_f}^{G' \subseteq G}$  in  $G' \subseteq G$  is:

$$\mathfrak{U}_{\pi_{v_0, v_f}^{G' \subseteq G}} = \prod_{(u, v) \in \pi_{v_0, v_f}^{G' \subseteq G}} \frac{\alpha}{\text{outdegree}_G(u)} = \alpha^{|\pi_{v_0, v_f}^{G' \subseteq G}|} \cdot \prod_{(u, v) \in \pi_{v_0, v_f}^{G' \subseteq G}} \frac{1}{\text{outdegree}_G(u)} \quad (2)$$

**Definition 3.** Given a node  $u$  and a node  $v$  in a graph  $G$ , the *conductance*  $\mathfrak{U}_{u, v}^{G' \subseteq G}$  from  $u$  to  $v$  in  $G'$  is the sum of the conductances of all paths in  $G'$  from  $u$  to  $v$ :

$$\mathfrak{U}_{u, v}^{G' \subseteq G} = \sum_{\pi_{u, v}^{G' \subseteq G}} \mathfrak{U}_{\pi_{u, v}^{G' \subseteq G}} \quad (3)$$

It is important to observe that there always exists a path from a node to itself, the empty path of length 0, whose conductance is exactly 1; thus in any graph the conductance of a node to itself is always at least 1. Another simple but crucial observation is that conductance between two nodes depends on, and is monotonically non-decreasing with, the subgraph  $G'$  on which it is computed, i.e. if  $G' \subseteq G'' \subseteq G$  then  $\mathfrak{U}_{u, v}^{G' \subseteq G} \leq \mathfrak{U}_{u, v}^{G'' \subseteq G}$ . To lighten the notation, throughout the rest of the paper, whenever  $G$  is obvious we shall simply write  $\text{outdegree}(u)$  in place of  $\text{outdegree}_G(u)$  and  $G'$  in place of  $G' \subseteq G$ .

**A “diffuse” formulation of the PageRank score.** Let  $v$  be the node whose PageRank score  $P(v)$  we want to estimate. Given a subgraph  $\bar{G} = (\bar{V}, \bar{E})$  of  $G = (V, E)$  such that  $v \in \bar{G}$ , denote by  $F(\bar{G}) = \{(u, w) \in E : u \notin \bar{V}, w \in \bar{V}\}$  its (incoming arc) *frontier*, i.e. the set of arcs of  $G$  not in  $\bar{G}$  but leading to nodes of  $\bar{G}$ . The following lemma (whose proof can be found in Appendix A.4), shows how one can express  $P(v)$  in terms of conductances to  $v$  in  $\bar{G}$ , of the PageRank scores of the nodes adjacent to the frontier of  $\bar{G}$ , and of the PageRank scores of the dangling nodes in  $G$  – assuming  $\bar{G}$  itself does not contain any dangling nodes except possibly  $v$ .

**Lemma 2.** *Given a graph  $G$ , a subgraph  $\bar{G} \subseteq G$ , and a node  $v \in \bar{G}$ , if  $\bar{G}$  contains no dangling nodes except possibly  $v$ , then  $P(v)$  equals:*

$$P^{\bar{G}}(v) = \mu_v^{\bar{G}} \cdot \left( \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \left( \frac{1-\alpha}{n} + \sum_{\substack{u \in G \setminus \{v\}: \\ \text{outdegree}(u)=0}} \alpha \frac{P(u)}{n} \right) + \sum_{(u,w) \in F(\bar{G})} \left( \alpha \frac{P(u)}{\text{outdegree}(u)} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) \right) \quad (4)$$

where  $\mu_v^{\bar{G}} = 1/(1 - \frac{\alpha}{n} \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}})$  if  $\text{outdegree}(v) = 0$ , and  $\mu_v^{\bar{G}} = 1$  otherwise.

For each  $u$  let  $x_u = P(u)$  if  $\text{outdegree}(u) = 0$  and  $x_u = P(u)/\text{outdegree}(u)$  if  $\text{outdegree}(u) > 0$ ; and let  $x_\emptyset = \sum_{u \in G \setminus \{v\}: \text{outdegree}(u)=0} P(u)$ . We can then re-write Equation 4 as:

$$P^{\bar{G}}(v) = c^{\bar{G}} \left( 1 + \frac{\alpha}{1-\alpha} \cdot x_\emptyset \right) + \sum_{u: (u,w) \in F(\bar{G})} c_u^{\bar{G}} \cdot x_u \quad (5)$$

where the coefficients  $c^{\bar{G}}$  and  $c_u^{\bar{G}}$  depend solely on the subgraph  $\bar{G}$ . Also, since  $P(v) = P^{\bar{G}}(v)$  for any  $\bar{G}$ , we can write  $P(v)$  as the weighted average of several  $P^{\bar{G}}(v)$  for different  $\bar{G}$ .

More precisely, consider  $m+1 \leq n$  subgraphs (of  $G$ )  $G_0, \dots, G_m$ , with  $G_0 = (\{v\}, (v, v) \cap E)$  and each subsequent  $G_i$  obtained by adding to  $G_{i-1}$  a new node  $u_i$  adjacent to its frontier as well as all arcs between  $u_i$  and  $G_{i-1} \cup \{u_i\}$ . More formally, if  $G = (V, E)$ , then  $G_i = (V_i, E_i)$  with  $V_i$  and  $E_i$  for  $i > 0$  defined respectively, given some  $u_i \in V \setminus V_{i-1}$  such that  $(u_i, u) \in F(G_{i-1})$ , as  $V_{i-1} \cup \{u_i\}$  and  $E \cap (V_i \times V_i)$ . If  $\beta_0^m, \dots, \beta_m^m$  are  $m+1$  non-negative weights such that  $\sum_{i=0}^m \beta_i^m = 1$ , we can then write:

$$P(v) = \sum_{i=0}^m \beta_i^m \cdot P^{G_i}(v) \quad (6)$$

Denote by  $j_i$  the smallest  $j$  such that there exists some  $u \in V_j$  with  $(u_i, u) \in E$ . From the definition of  $E_i$ , we have that  $(u_i, u) \in F(G_j) \forall j : j_i \leq j < i$  – i.e.  $(u_i, u)$  remains in the frontier until  $u_i$  is added to the node set. Also, note that no arc originating from  $u_i$  is ever in  $F(G_j)$  for any  $j \geq i$ . We then call the set of indices  $j$  for which  $u_i$  has some outgoing arc in  $F(G_j)$ , and therefore appears in the summation on the right-hand side of Equation 5, the *frontier interval*  $\mathcal{I}_i = [j_i, i-1]$ ; and we say that  $u_i$  *appears on the frontier* at  $j_i$ , and *crosses the frontier* at  $i$ . Obviously, no dangling node ever appears on the frontier, and therefore  $G_i$  cannot contain any dangling node besides (possibly)  $v$ . We can thus combine Equations 5 and 6 and write:

$$P(v) = \sum_{i=0}^m \beta_i^m \cdot c^{G_i} \left( 1 + \frac{\alpha}{1-\alpha} \cdot x_\emptyset \right) + \sum_{i=1}^m \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} x_{u_i} + \sum_{i: (u_i, w) \in F(G_m)} \sum_{j=j_i}^m \beta_j^m \cdot c_i^{G_j} x_{u_i} \quad (7)$$

The right-hand side of Equation 7 has three terms. The first depends on the topology of  $G_0, \dots, G_m$  and on the PageRank scores of the dangling nodes of  $G \setminus \{v\}$ . The second is a weighted sum of all  $x_u$  such that  $u$  has appeared on, and has crossed, the frontier of some  $G_i$  with  $i \leq m$  – with the weights depending solely on the topology of  $G_0, \dots, G_m$ . Note that this



sum is zero for  $i = 0$ . The third term is a weighted sum of all  $x_u$  such that  $u$  has appeared on the frontier of some  $G_i$  with  $i \leq m$ , and has remained on the frontier of  $G_i, \dots, G_m$  – again with the weights depending solely on the topology of  $G_0, \dots, G_m$ .

Note that Equation 7 provides a formula for  $P(v)$  that depends solely on the topology of  $G_0, \dots, G_m$ , on the PageRank scores and outdegrees of nodes in  $G_m$  or with an outgoing arc in  $F(G_m)$  (the identity of the latter can be obtained by invoking  $Neighbourhood(\cdot)$  on every node of  $G_m$ ), and on the PageRank scores of the dangling nodes in  $G \setminus \{v\}$ . Consider  $\bar{n}$  invocations of  $SampleNode()$  and denote by  $\chi_i^h$  the indicator variable of the event that the  $h^{th}$  invocation returns  $u_i$ , and by  $\chi_\emptyset^h = \sum_{u_i \in G \setminus \{v\}: outdegree(u_i)=0} \chi_i^h$  that of the event that the  $h^{th}$  invocation returns any of the dangling nodes in  $G \setminus \{v\}$ . Remembering that  $P(u_i) = E[\chi_i^h]$ , we have that  $P(v)$  equals the expectation of the random variable  $p_{G_m}^{\bar{n}}$  defined as:

$$p_{G_m}^{\bar{n}} = \frac{1}{\bar{n}} \sum_{h=1}^{\bar{n}} \left( \sum_{i=0}^m \beta_i^m \cdot c^{G_i} \left( 1 + \frac{\alpha}{1-\alpha} \cdot \chi_\emptyset^h \right) + \sum_{i=1}^m \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} \frac{\chi_i^h}{outdegree(u_i)} \right. \\ \left. + \sum_{i:(u_i, w) \in F(G_m)} \sum_{j=j_i}^m \beta_j^m \cdot c_i^{G_j} \frac{\chi_i^h}{outdegree(u_i)} \right) \quad (8)$$

It is important to observe that one can obtain all information necessary to take a sample of  $p_{G_m}^{\bar{n}}$  by invoking  $SampleNode()$   $\bar{n}$  times and  $Neighbourhood(\cdot)$  at most  $\bar{n} + (m+1)$  times (on the  $m+1$  nodes of  $G_m$ , and on those nodes returned by  $SampleNode()$ , which are at most  $\bar{n}$ , to learn their outdegrees). This is true even though the nodes involved in Equation 7 (and in particular those in the third term of its right-hand side) may be far more than  $\bar{n} + (m+1)$ : the outdegree of  $u_i$  is irrelevant in Equation 8 if  $\chi_i^h$  is 0 for all  $h$ .

**Probability bounds for large deviations.** Having established that  $P(v) = E[p_{G_m}^{\bar{n}}]$ , and that a sample of  $p_{G_m}^{\bar{n}}$  requires at most  $\bar{n}$  invocations of  $SampleNode()$  and  $\bar{n} + (m+1)$  of  $Neighbourhood(\cdot)$ , we must now bound the probability that such a sample is not within a factor  $(1 \pm \epsilon)$  of its expectation. By Equation 8 we can write  $p_{G_m}^{\bar{n}} = q_{G_m}^{\bar{n}} + s_{G_m}^{\bar{n}}$ , where:

$$q_{G_m}^{\bar{n}} = \frac{1}{\bar{n}} \sum_{h=1}^{\bar{n}} \left( \sum_{i=0}^m \beta_i^m \cdot c^{G_i} \left( 1 + \frac{\alpha}{1-\alpha} \cdot \chi_\emptyset^h \right) \right) \quad (9)$$

$$s_{G_m}^{\bar{n}} = \frac{1}{\bar{n}} \sum_{h=1}^{\bar{n}} \left( \sum_{i=1}^m \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} \frac{\chi_i^h}{outdegree(u_i)} + \sum_{i:(u_i, w) \in F(G_m)} \sum_{j=j_i}^m \beta_j^m \cdot c_i^{G_j} \frac{\chi_i^h}{outdegree(u_i)} \right) \quad (10)$$

so if  $q_{G_m}^{\bar{n}}$  and  $s_{G_m}^{\bar{n}}$  both fall within a factor  $(1 \pm \epsilon)$  of their expectation, then  $p_{G_m}^{\bar{n}}$  does too.

**Dealing with dangling nodes.** Clearly,  $q_{G_m}^{\bar{n}} = \frac{1}{\bar{n}} \left( \sum_{i=0}^m \beta_i^m \cdot c^{G_i} \right) \left( \bar{n} + \frac{\alpha}{1-\alpha} \sum_{h=1}^{\bar{n}} \chi_\emptyset^h \right)$  falls within a factor  $(1 \pm \epsilon)$  of its expectation if and only if  $\bar{n} + \frac{\alpha}{1-\alpha} \sum_{h=1}^{\bar{n}} \chi_\emptyset^h$  does. Note that, if there are no dangling nodes in  $G \setminus \{v\}$ , then  $\chi_\emptyset^h = 0$  for every  $h$ , yielding  $q_{G_m}^{\bar{n}} = E[q_{G_m}^{\bar{n}}] = \sum_{i=0}^m \beta_i^m c^{G_i}$ . Otherwise, given that  $E[\sum_{h=1}^{\bar{n}} \chi_\emptyset^h] = \bar{n} x_\emptyset$ , we obtain:

$$Pr \left[ \frac{q_{G_m}^{\bar{n}}}{E[q_{G_m}^{\bar{n}}]} \notin [1 - \epsilon, 1 + \epsilon] \right] \\ = Pr \left[ \bar{n} + \frac{\alpha}{1-\alpha} \sum_{h=1}^{\bar{n}} \chi_\emptyset^h \notin \left[ (1 - \epsilon) \left( \bar{n} + \frac{\alpha}{1-\alpha} \bar{n} x_\emptyset \right), (1 + \epsilon) \left( \bar{n} + \frac{\alpha}{1-\alpha} \bar{n} x_\emptyset \right) \right] \right] \quad (11)$$

$$\begin{aligned}
&=Pr \left[ \sum_{h=1}^{\bar{n}} \chi_{\emptyset}^h \notin \left[ (1-\epsilon)\bar{n}x_{\emptyset} - \epsilon\bar{n}\frac{1-\alpha}{\alpha}, (1+\epsilon)\bar{n}x_{\emptyset} + \epsilon\bar{n}\frac{1-\alpha}{\alpha} \right] \right] \\
&=Pr \left[ \sum_{h=1}^{\bar{n}} \chi_{\emptyset}^h \notin \left[ \left(1-\epsilon\left(1+\frac{1-\alpha}{\alpha x_{\emptyset}}\right)\right)\bar{n}x_{\emptyset}, \left(1+\epsilon\left(1+\frac{1-\alpha}{\alpha x_{\emptyset}}\right)\right)\bar{n}x_{\emptyset} \right] \right]
\end{aligned}$$

Since different invocations of *SampleNode()* are independent, the indicator variables  $\chi_{\emptyset}^h$  are non-positively correlated, and a simple probability bound (see Appendix A.1) gives for all  $\epsilon \in [0, 1]$ :

$$Pr \left[ \frac{q_{G_m}^{\bar{n}}}{E[q_{G_m}^{\bar{n}}]} \notin [1-\epsilon, 1+\epsilon] \right] < 2e^{-\bar{n}x_{\emptyset}\epsilon^2\left(1+\frac{1-\alpha}{\alpha x_{\emptyset}}\right)^2/3} < 2e^{-\bar{n}\frac{\epsilon^2}{3x_{\emptyset}}\left(\frac{1-\alpha}{\alpha}\right)^2} \leq 2e^{-\bar{n}\frac{\epsilon^2}{3}\left(\frac{1-\alpha}{\alpha}\right)^2} \quad (12)$$

which can be made arbitrarily smaller than  $\epsilon$  for  $\bar{n} \in O(1)$  (the last inequality follows from the fact that  $x_{\emptyset} = E[\chi_{\emptyset}^h] \leq 1$ ).

**Balancing the coefficients.** To bound the probability that  $s_{G_m}^{\bar{n}}$  does not fall within a factor  $(1 \pm \epsilon)$  of its expectation, let us examine how the three terms in the right-hand side of Equation 7 change if we add one more subgraph  $G_{m+1}$  (obtained by adding to  $G_m$  a new node  $u_{m+1}$  and all arcs between  $u_{m+1}$  and  $G_m \cup \{u_{m+1}\}$ ) with weight  $\beta_{m+1}^{m+1}$ , assuming all other weights are decreased by a multiplicative factor  $(1 - \beta_{m+1}^{m+1})$ , so that:

$$\beta_i^{m+1} = (1 - \beta_{m+1}^{m+1})\beta_i^m \quad 0 \leq i \leq m \quad (13)$$

Note that by Equation 13, the set of coefficients  $\{\beta_j^i : 0 \leq j = i \leq m\}$  uniquely determines its superset  $\{\beta_j^i : 0 \leq j \leq i \leq m\}$ . We can then re-write Equation 7 for  $m+1$  as:

$$P(v) = \sum_{i=0}^{m+1} \beta_i^{m+1} \cdot c^{G_i} \left(1 + \frac{\alpha}{1-\alpha} \cdot x_{\emptyset}\right) + \sum_{i=1}^{m+1} \sum_{j=j_i}^{i-1} \beta_j^{m+1} \cdot c_i^{G_j} x_{u_i} + \sum_{i:(u_i, w) \in F(G_{m+1})} \sum_{j=j_i}^{m+1} \beta_j^{m+1} \cdot c_i^{G_j} x_{u_i} \quad (14)$$

Let us examine the fundamental changes between Equation 7 and Equation 14. The first summation (containing no  $x_u$ ) changes according to the topology of  $G_{m+1}$ . The second summation “steals” from the third the term  $\sum_{j=j_{m+1}}^m \beta_j^m \cdot c_{m+1}^{G_j} x_{u_{m+1}}$ , as  $u_{m+1}$  crosses the frontier; and is then multiplied by  $(1 - \beta_{m+1}^{m+1})$  – crucially, this reduces the weights of all  $x_u$  by exactly the same factor. The third summation “loses” to the second the term  $\sum_{j=j_{m+1}}^m \beta_j^m \cdot c_{m+1}^{G_j} x_{u_{m+1}}$  and conversely gains zero or more terms in the form  $\sum_{j=m+1}^{m+1} \beta_j^{m+1} \cdot c_{\bar{h}}^{G_j} x_{u_{\bar{h}}}$  corresponding to each node  $u_{\bar{h}}$  that, having an outgoing arc insisting on  $u_{m+1}$ , appears on the frontier as  $u_{m+1}$  crosses it. In addition, every other node  $u_i$  that had already appeared on, but not crossed, the frontier gains a new term  $\beta_j^{m+1} \cdot c_i^{G_{m+1}} x_{u_i}$  proportional to  $\beta_{m+1}^{m+1}$  in the corresponding inner sum; thus all those terms (whether corresponding to “new” or “old” frontier nodes) can be made arbitrarily small for a sufficiently small  $\beta_{m+1}^{m+1}$  (and are in fact equal to 0 if  $\beta_{m+1}^{m+1} = 0$ ).

Associate to each node  $u$  of  $G$  an arbitrary positive multiplier  $\frac{1}{\gamma_u}$ . We now show how to construct the sequence of nodes  $u_i$  that will form  $G_1, G_2, \dots$ , and the corresponding sequence of weights  $\beta_i^i$  (note that by definition  $\beta_0^0 = 1$ ) in such a way that, immediately after  $u_m$  is added, for  $1 \leq i \leq m$  the coefficient  $\sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j}$  multiplying each  $x_{u_i}$  satisfies simultaneously the following three conditions:

$$\forall i' : 1 \leq i' \leq m : \quad \frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} = \frac{1}{\gamma_{u_{i'}}} \cdot \sum_{j=j_{i'}}^{i'-1} \beta_j^m \cdot c_{i'}^{G_j} \quad (15)$$

$$\forall i' : i' > m : \quad \frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} \geq \frac{1}{\gamma_{u_{i'}}} \cdot \sum_{j=j_{i'}}^m \beta_j^m \cdot c_{i'}^{G_j} \quad (16)$$

$$F(G_m) \neq \emptyset \implies \exists i' > m : \quad \frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j} = \frac{1}{\gamma_{u_{i'}}} \cdot \sum_{j=j_{i'}}^m \beta_j^m \cdot c_{i'}^{G_j} \quad (17)$$

Roughly speaking, we want the weighted coefficient of a node  $u$  that has just appeared on the frontier to start at 0 (meaning at that point it does not violate Equation 16) and progressively grow until it reaches the weighted coefficients of those nodes that have already crossed the frontier. At that point it satisfies Equation 17 and  $u$  becomes next-in-line for crossing the frontier itself. When it does, it satisfies Equation 15 – and it keeps doing so from then on, since the weighted coefficients of those nodes that have crossed the frontier all decrease at the same rate. For a formal construction of the sequence of nodes  $u_i$  and of the corresponding sequence of weights  $\beta_i^i$  see Appendix A.5.

Before proceeding, let us briefly consider the condition  $F(G_m) \neq \emptyset$  in Equation 17. Obviously, if  $F(G_m)$  becomes empty, we can no longer expand  $G_m$  into  $G_{m+1}$  – but we can then accurately estimate  $P(v)$  from Equation 5, as the first term on the right-hand side disappears and the second can be accurately estimated with  $O(1)$  queries once an accurate estimate of  $n$  is known (see above). We shall then henceforth assume without loss of generality that  $F(G_m) \neq \emptyset$ .

**A “good” linear combination leads to low variance.** Recall (see Appendix A.1) that the probability that a sum with expectation  $\mu$  of non-positively correlated random variables with support in  $[0, 1]$  is not within a factor  $(1 \pm \epsilon)$  of  $\mu$  is, for any positive  $\epsilon \leq 1$ , less than  $2e^{-\frac{\epsilon^2 \mu}{3}}$ . Let  $\ell$  be index of the largest coefficient of any  $\chi_i^h$  in Equation 10,

$$\ell = \arg \max_{i: u_i \in G_m \vee (u_i, w) \in F(G_m)} \left( \frac{1}{\text{outdegree}(u_i)} \sum_{j=j_i}^{\min(m, i-1)} \beta_j^m \cdot c_i^{G_j} \right) \quad (18)$$

and consider the random variable  $r_{G_m}^{\bar{n}}$  defined as:

$$\begin{aligned} r_{G_m}^{\bar{n}} &= \bar{n} \cdot \frac{s_{G_m}^{\bar{n}}}{\frac{1}{\text{outdegree}(u_\ell)} \sum_{j=j_\ell}^{\min(m, \ell-1)} \beta_j^m \cdot c_\ell^{G_j}} \\ &= \sum_{h=1}^{\bar{n}} \sum_{i=1}^m \frac{\frac{1}{\text{outdegree}(u_i)} \sum_{j=j_i}^{i-1} \beta_j^m \cdot c_i^{G_j}}{\frac{1}{\text{outdegree}(u_\ell)} \sum_{j=j_\ell}^{\min(m, \ell-1)} \beta_j^m \cdot c_\ell^{G_j}} \cdot \chi_i^h \\ &\quad + \sum_{h=1}^{\bar{n}} \sum_{i: (u_i, w) \in F(G_m)} \frac{\frac{1}{\text{outdegree}(u_i)} \sum_{j=j_i}^m \beta_j^m \cdot c_i^{G_j}}{\frac{1}{\text{outdegree}(u_\ell)} \sum_{j=j_\ell}^{\min(m, \ell-1)} \beta_j^m \cdot c_\ell^{G_j}} \cdot \chi_i^h \end{aligned} \quad (19)$$

From Equation 19 we have  $s_{G_m}^{\bar{n}} = c_1^m \cdot r_{G_m}^{\bar{n}}$  with  $c_1^m = \frac{\sum_{j=j_\ell}^{\min(m, \ell-1)} \beta_j^m \cdot c_\ell^{G_j}}{\bar{n} \cdot \text{outdegree}(u_\ell)} > 0$ , and thus for all  $\epsilon \geq 0$  the probability that  $s_{G_m}^{\bar{n}}$  is within a multiplicative factor  $(1 \pm \epsilon)$  of its expectation equals the probability that  $r_{G_m}^{\bar{n}}$  is. Note that  $r_{G_m}^{\bar{n}}$  is a weighted sum of the indicator variables  $\chi_i^h$  (of the event that the  $h^{\text{th}}$  invocation of *SampleNode*() returns  $u_i$ ) with non-negative weights no larger than 1 (by the definition of  $\ell$  in Equation 18). Since outcomes of different samples are

independent, while different outcomes of the same sample are mutually exclusive, a probability bound on the sum of non-positively correlated random variables (see Appendix A.1) yields:

$$\forall \epsilon \in [0, 1] \quad \Pr \left[ \frac{s_{G_m}^{\bar{n}}}{E[s_{G_m}^{\bar{n}}]} \notin \left[ (1 - \epsilon), (1 + \epsilon) \right] \right] \leq 2e^{-\frac{\epsilon^2 E[r_{G_m}^{\bar{n}}]}{3}} \quad (20)$$

We now show how one can (probabilistically) select with  $O(n^{\frac{2}{3}} \sqrt[3]{\log(n)})$  queries  $m$ ,  $\bar{n}$  and  $\gamma_{u_1}, \dots, \gamma_{u_m}$  so as to make  $\epsilon^2 \cdot E[r_{G_m}^{\bar{n}}]$  arbitrarily larger than  $\log(\frac{1}{\epsilon})$ ; from Equation 20, this makes arbitrarily smaller than  $\epsilon$  the probability of erring in our estimate of  $E[s_{G_m}^{\bar{n}}]$  by more than a factor  $(1 \pm \epsilon)$ .

Note that, from Equation 15, the coefficient multiplying each  $\chi_i^h$  for  $i \leq m$  in Equation 8 is proportional to the ratio  $\frac{\gamma_{u_i}}{\text{outdegree}(u_i)}$ ; and from Equations 16 and 18,  $\frac{\gamma_{u_\ell}}{\text{outdegree}(u_\ell)}$  is the largest such ratio for all  $u_i$  that are either in  $G_m$  or with an outgoing arc in  $F(G_m)$ . Thus, considering just the contribution to  $E[r_{G_m}^{\bar{n}}]$  of the coefficients of  $\chi_i^h$  for  $i \leq m$ , we can write:

$$E[r_{G_m}^{\bar{n}}] \geq \bar{n} \cdot \frac{\sum_{i=1}^m \gamma_{u_i} / \text{outdegree}(u_i)}{\gamma_{u_\ell} / \text{outdegree}(u_\ell)} \cdot E[\chi_i^h] \quad (21)$$

Remembering that  $E[\chi_i^h] = P(u_i) \geq \frac{1-\alpha}{n}$ , we then have:

$$E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \cdot \frac{\bar{n}}{n} \cdot \frac{\sum_{i=1}^m \gamma_{u_i} / \text{outdegree}(u_i)}{\gamma_{u_\ell} / \text{outdegree}(u_\ell)} \quad (22)$$

If we knew the outdegree of every  $u_i$ , setting  $\gamma_{u_i} = \text{outdegree}(u_i)$ , for  $\bar{n} = m = \Theta(n^{\frac{1}{2}})$  we would have  $E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \frac{\bar{n} \cdot m}{n} = \Theta(1)$ , and the constant involved could be made arbitrarily large for a sufficiently large  $\bar{n} = m$  (see Appendix A.6 for a discussion on this point). Unfortunately, in general we do not know the outdegree of *all*  $u_i$  with an outgoing arc in  $F(G(m))$  (and these could be  $\Omega(n)$ , i.e. too many to invoke on each of them  $\text{Neighbourhood}(\cdot)$ ); and although by the time  $G_m$  has been constructed we know the outdegree of all its nodes, we must choose  $\gamma_{u_i}$  *before* we choose  $u_i$  since knowledge of  $\gamma_{u_i}$  is instrumental in the choice. This may cause an “imbalance” in the coefficients of different  $\chi_i^h$  in Equation 8, and  $E[r_{G_m}^{\bar{n}}]$  may well fall to  $o(1)$  if the largest coefficient is much larger than the average. Note that this is not an artifact of the analysis: giving excessive weight to a particular node  $u_i$  in the construction of  $G_m$  makes  $p_{G_m}^{\bar{n}}$  behave more like a sum of  $\bar{n}$  random variables than as a sum of  $\bar{n} \cdot m$ , increasing its variance.

**Guessing outdegrees.** We can contain the imbalance in the coefficients by “guessing” the outdegree of every node that will eventually appear on the frontier of  $G_m$  and setting  $\gamma_{u_i}$  accordingly, *before* the construction of  $G_1, \dots, G_m$  even begins (i.e. even before selecting  $u_1$ ).

Invoke *RandomNode()* a total of  $\bar{n}$  times, and for the  $h^{\text{th}}$  node  $v_h$  returned consider the set  $A_h$  of its parents (yielded by a single invocation of  $\text{Neighbourhood}(v_h)$ ). Then, for each node  $u$ , let  $\psi_u^h$  be the indicator variable of the event  $u \in A_h$  and consider the random variable:

$$g(u) = \frac{n}{\bar{n}} \left( c \log(n) + \sum_{h=1}^{\bar{n}} \psi_u^h \right) \quad (23)$$

for some positive constant  $c$  to be defined later. Since  $E[\psi_u^h] = \frac{\text{outdegree}(u)}{n}$  and thus  $E[\frac{n}{\bar{n}} \cdot \sum_{h=1}^{\bar{n}} \psi_u^h] = \text{outdegree}(u)$ , we are then setting  $g(u)$  to the estimate of  $\text{outdegree}(u)$  obtained from the  $\psi_u^h$ , plus a fixed quantity  $\frac{cn \log(n)}{\bar{n}}$ . During the construction of  $G_1, \dots, G_m$ , whenever a node  $u_i$  appears on the frontier, we simply set  $\gamma_{u_i} = g(u_i)$ .

A careful analysis of the rightmost term in the product forming the right-end side of Equation 22 (see Appendix A.7) yields with probability greater than  $1 - n^{-\Theta(c)}$  for all sufficiently

large  $c$  (such that  $n^{-\frac{c}{8\log_e(2)}} \cdot n = n^{-\Theta(c)}$ ):

$$E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \frac{\bar{n}}{n} \cdot \frac{m/2}{2cn \log(n)/\bar{n}} = (1 - \alpha) \frac{m\bar{n}^2}{4cn^2 \log(n)} \quad (24)$$

and the last term is at least  $\frac{1-\alpha}{4}c^2$  for  $m = \bar{n} = c \cdot n^{\frac{2}{3}} \sqrt[3]{\log(n)}$ . Plugging this value into Equation 20 and choosing a sufficiently large  $c \in \Theta\left(\frac{1}{\epsilon} \sqrt{\log(\frac{1}{\epsilon})}\right)$  yields a probability arbitrarily smaller than  $\epsilon$  that  $s_{G_m}^{\bar{n}}$  is not within a factor  $(1 \pm \epsilon)$  of its expectation, provided one has a “good” set of  $\gamma_{u_i}$ . Recall that the probability of obtaining a “bad” set of  $\gamma_{u_i}$  is at most  $n^{-\Theta(c)}$  (see Appendix A.7) and the probability of obtaining an estimate of  $q_{G_m}^{\bar{n}}$  not within a factor  $(1 \pm \epsilon)$  of its expectation can be made arbitrarily smaller than  $\epsilon$  (see Equation 12) – at least as long as one has a “good” estimate of  $n$ , which fails to happen with probability arbitrarily smaller than  $\epsilon$ . Then, by a simple union bound, the probability of obtaining an estimate of  $P(v) = E[p_{G_m}^{\bar{n}}] = E[q_{G_m}^{\bar{n}}] + E[s_{G_m}^{\bar{n}}]$  that is not within a factor  $(1 \pm \epsilon)$  of its expectation can be made (arbitrarily) smaller than  $\epsilon$  with  $O\left(\frac{1}{\epsilon} \sqrt{\log(\frac{1}{\epsilon})} \cdot n^{\frac{2}{3}} \sqrt[3]{\log(n)}\right)$  queries, proving Theorem 1.

**Knowing outdegrees.** Remember that, from the observation immediately following Equation 22, if one can somehow learn for each node  $u$  involved in the construction of  $G_m$  (whether belonging to it or with an outgoing arc in its frontier) *before the construction of  $G_1, \dots, G_m$  begins* that  $\text{outdegree}(u)$  falls within a known interval  $[\text{mindeg}(u), \text{maxdeg}(u)]$ , one no longer needs to estimate  $\text{outdegree}(u)$  through  $\text{RandomNode}()$  calls and the number of necessary queries may asymptotically decrease if  $\max_u \frac{\text{maxdeg}(u)}{\text{mindeg}(u)}$  is sufficiently small.

Such information on node outdegrees could be available, for example, if using a more powerful version of the  $\text{Neighbourhood}(\cdot)$  primitive, providing outdegree information for each node returned. An alternative scenario in which sufficient outdegree information is available to break the bounds of Theorem 2 is when dealing with graphs with intrinsically restricted outdegrees – like those modelling some social networks or citation graphs. If for every node  $u$  we have that  $\frac{\text{maxdeg}(u)}{\text{mindeg}(u)} = o(n^{\frac{1}{3}})$  (e.g. if  $\text{maxdeg}(u) = o(n^{\frac{1}{3}})$ ) then the number of queries necessary decreases to  $O\left(\frac{1}{\epsilon} \sqrt{\log(\frac{1}{\epsilon})} \cdot n^{\frac{1}{2}} \sqrt{\max_u \left(\frac{\text{maxdeg}(u)}{\text{mindeg}(u)}\right)}\right)$ , thus proving Theorem 1A.

## 4 Conclusions

Somewhat surprisingly, the  $\Omega(n)$  query complexity bound that holds when restricted to local queries can be broken by simply adding  $\text{RandomNode}()$  queries (that just return a node chosen uniformly at random). The combination of  $\text{Neighbourhood}(\cdot)$  and  $\text{RandomNode}()$  queries guarantees a query complexity of  $\tilde{O}(n^{\frac{2}{3}})$  that is essentially optimal for the general case of arbitrary nodes in arbitrary graphs using any type of natural queries.

To break through the new  $\tilde{O}(n^{\frac{2}{3}})$  barrier established by this paper, one must then inevitably resort to analysing graphs or nodes with “special” properties. As noted, this may not be overly restrictive in many cases of practical interest: e.g. when the maximum outdegree of the graph is polylogarithmic, which accurately models many social contexts, query complexity drops to  $\tilde{O}(n^{\frac{1}{2}})$  even with the same techniques we use for the general case. Characterizing these properties seems then an obvious line of future research.

Another interesting direction of future research is understanding whether the combination of global sketching and local exploration which lies at the heart of our algorithm can be applied to other local graph problems. Particularly promising candidates seem to be other Markov Chain Monte Carlo computations [22]: when can one, adapting our technique, obtain a  $(1 \pm \epsilon)$  approximation of the stationary probability of any one state in a Markov chain by exploring only a vanishing portion of the state space?

## A Appendix

### A.1 Probability bounds

In this work we repeatedly use the following result, often cited as the Angluin-Valiant bound [4] – which is a simpler version, “tailored” for sums whose expectation is much smaller than the number of variables, of the Chernoff-Hoeffding bound written in terms of relative entropy.

**Lemma 3.** *Let  $X$  be the sum of  $n$  random variables  $X_1, \dots, X_n$  in  $[0, 1]$  that are non-positively correlated (i.e.  $E[X_1 \dots X_n] \leq \prod_{i=1}^n E[X_i]$ ). Then, for any  $\Delta > 0$ , we have:*

$$\Pr[X < (1 - \Delta)E[X]] < e^{-\frac{\Delta^2}{2}E[X]} \quad (25)$$

$$\Pr[X > (1 + \Delta)E[X]] < \left( \frac{e^\Delta}{(1 + \Delta)^{1+\Delta}} \right)^{E[X]} \leq e^{-\frac{\Delta^2}{2+\Delta}E[X]} \quad (26)$$

Note that Lemma 3 applies if  $X_1, \dots, X_n$  are indicator variables of mutually disjoint events, or if they can be partitioned into independent families  $\{X_1, \dots, X_{i_1}\}, \{X_{i_1+1}, \dots, X_{i_2}\}, \dots$  of such variables (since  $E[X_1 \dots X_n] = E[X_1, \dots, X_{i_1}] \cdot E[X_{i_1+1}, \dots, X_{i_2}] \cdot \dots = 0 \cdot 0 \dots = 0$ ).

### A.2 Proof of Theorem 2A

We analyse separately, but with similar techniques, the two cases  $\gamma(n) > n^{\frac{1}{3}}$  and  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$ , assembling the general lower bound at the end. We will use the following equivalent formulation of PageRank, which is easier to compute explicitly on the graphs appearing in this proof:

$$P(v) = \frac{1 - \alpha}{n} \sum_{\tau=0}^{+\infty} \alpha^\tau \left( \sum_{z \in G} \left( \sum_{\pi_{z,v}: |\pi_{z,v}|=\tau} \left( \prod_{(w,w') \in \pi_{z,v}} \frac{1}{\text{outdegree}(w)} \right) \right) \right) \quad (27)$$

where a path  $\pi_{z,v}$  is a sequence of zero or more arcs leading from  $z$  to  $v$ , and  $|\pi_{z,v}|$  is the number of such arcs. In the case  $\text{outdegree}(v) = 1$  and its arc forms a self loop, we simplify the computation of  $P(v)$  as in [14] by disregarding that self loop (and thus all the paths it belongs to), applying Equation 27, and then multiplying the result by  $\frac{1}{1-\alpha}$ . To keep the proof simple, we assume that  $f(n)$  and  $\gamma(n)$  are respectively in  $\Theta(f(n_0))$  and  $\Theta(\gamma(n_0))$  whenever  $n = \Theta(n_0)$ , and show how to remove this hypothesis at the end.

**Case  $\gamma(n) > n^{\frac{1}{3}}$ .** We proceed as follows: first, for any  $n^{-2/3} \leq f(n) \leq 1$  we exhibit a family  $\mathcal{F}_n$  of arbitrarily large, isomorphic graphs on  $n$  nodes whose generic element has maximum outdegree  $O(\gamma(n))$  and contains two nodes  $u, v$  with PageRank scores in  $\Theta(f(n))$  but not within a factor  $(1 + \eta)$  of each other. We prove that any algorithm, on some element in  $\mathcal{F}_n$ , must perform  $\Omega(\frac{1}{f(n)})$  queries in expectation to decide which among  $u$  and  $v$  has the higher score with probability  $\frac{1}{2} + \Omega(1)$ . We then show how to adapt  $\mathcal{F}_n$  for  $\frac{1}{n} \leq f(n) < n^{-2/3}$  while bringing the above expectation to  $\Omega(n^{2/3})$ , thus obtaining a lower bound of  $\Omega(\min(\frac{1}{f(n)}, n^{2/3}))$  for any  $\frac{1}{n} \leq f(n) \leq 1$ .

To build a generic element  $G$  of  $\mathcal{F}_n$  for  $n^{-2/3} \leq f(n) \leq 1$ , consider an arbitrary positive integer  $n_0$  (a rough approximation of the final number of nodes in  $G$ ). The nodes of  $G$  (see Figure 2) can be divided into three levels. Level 0 consists of 2 nodes  $u, v$  and  $2 \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil \left\lceil \frac{1}{\sqrt{f(n_0)}} \right\rceil$  nodes  $w_0^0, w_0^1, \dots$ ; each node in this level has (only) a self loop. Level 1 consists of 2 nodes  $s_u, s_v$  and  $2 \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$  nodes  $w_1^0, w_1^1, \dots$ . The sole outgoing arcs of  $s_u$  and  $s_v$  are, respectively,  $(s_u, u)$  and  $(s_v, v)$ ; while  $(w_1^j, u)$  is an arc for  $0 \leq j < \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$ , and  $(w_1^j, v)$  is

The diagram illustrates a hierarchical structure with three levels:

- level 2:** Contains nodes  $w_2^0$ ,  $w_2^1$ , ..., and a shaded node  $s_u$ .
- level 1:** Contains nodes  $w_1^0$ ,  $w_1^1$ , ..., and a shaded node  $s_v$ .
- level 0:** Contains nodes  $w_0^0$ ,  $w_0^1$ , ..., and nodes  $u$  and  $v$ .

Arrows indicate connections between levels. Dashed lines represent multiple connections. Self-loops are shown on nodes  $u$  and  $v$ .

The number of nodes in  $G$  is  $n = 4 + 2 \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil \left\lceil \frac{1}{\sqrt{f(n_0)}} \right\rceil + 2 \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil + \left\lceil \frac{cn_0 f(n_0)}{\alpha^2} \right\rceil$ , which is a  $\Theta(n_0)$  between  $\frac{2n_0}{\alpha}$  and  $\frac{17+c\eta}{\alpha^2}n_0$ ; therefore, one can make  $G$  arbitrarily large by increasing  $n_0$ . The maximum outdegree of the graph is  $\left\lceil \frac{1}{\sqrt{f(n_0)}} \right\rceil = \Theta\left(\frac{1}{\sqrt{f(n_0)}}\right)$ , and since  $f(n) \geq n^{-\frac{2}{3}}$ , this is an  $O(n^{\frac{1}{3}}) \subseteq O(\gamma(n))$ . The PageRank scores of  $u$  and  $v$  are:

$$P(v) = \frac{1}{n} \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil}{1 + \left\lceil \frac{1}{\sqrt{f(n_0)}} \right\rceil} \right) \right) \quad (29)$$

$$\frac{P(u) - P(v)}{P(v)} = \frac{\frac{1}{n}\alpha^2 \left\lceil \frac{c\eta n_0 f(n_0)}{\alpha^2} \right\rceil}{\frac{1}{n} \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil}{1 + \frac{1}{\sqrt{f(n_0)}}} \right) \right)} \geq \frac{\frac{1}{n} c\eta n_0 f(n_0)}{\frac{1}{n} \left( 1 + \alpha + \frac{2n_0 \sqrt{f(n_0)}}{1 + \sqrt{f(n_0)}} \right)} = \frac{c\eta}{\frac{1+\alpha}{n_0 f(n_0)} + 2} \quad (30)$$

14

Consider now a generic (Monte Carlo) algorithm that, using only natural exploration queries, must decide which among  $u$  and  $v$  has the higher PageRank score in a graph drawn uniformly at random from  $\mathcal{F}_n$  (the ids of  $u$  and  $v$  are given in input in random order). Note that the information returned by global queries is independent from their order relative to local queries, since the output of a global query does not depend on the output of previous local queries. We can then simplify the analysis by ideally grouping together all the global queries in a first phase, followed by all the local queries in a second phase, with each phase sporting at most  $q$  queries if the algorithm performs at most  $q$  queries overall.

Let us consider the first phase. We reinforce the algorithm with an oracle that operates as follows: when a global query outputs the id of a node among  $s_u, s_v$  or  $w_2^0, w_2^1, \dots$ , it lets the algorithm immediately return the correct ranking of the input nodes; and when a global query outputs the id of a level-0 node not in  $\{u, v\}$ , it reveals both the id of its unique parent (which can thus be marked as neither  $s_u$  nor  $s_v$ ) and the ids of all its siblings (revealing all the information that would be given by a  $Neighbourhood(\cdot)$  on their parent). Since the graph is drawn uniformly at random from  $\mathcal{F}_n$ , each single global query returns the id of a node among  $s_u, s_v$  or  $w_2^0, w_2^1, \dots$  with probability  $\left(2 + \left\lceil \frac{cn_0 f(n_0)}{\alpha^2} \right\rceil\right)/n = O(f(n))$ ; hence, the probability that some global query in the first phase returns such an id is  $O(q \cdot f(n))$ . If this does not happen, then the algorithm possesses no information about which one of the input nodes has the higher PageRank score: conditioned on the fact of *not* discovering any of the above nodes, each of the possible output sequences of the global queries has exactly the same probability.

Let us turn to the second phase, conditioned on the event that the first phase ended with the algorithm having no information about which input node has the higher score. Note that a single invocation of  $Neighbourhood(\cdot)$  yields sufficient information for the algorithm to emulate, without any further query invocation, any other type of local query; we can then assume that the algorithm employs only  $Neighbourhood(\cdot)$  queries, and slightly reinforce it by allowing two  $Neighbourhood(\cdot)$  invocations on the two input nodes at no cost before the phase begins. At this point, the algorithm knows the ids of all level-1 nodes; and, possibly, the ids of some level-0 nodes discovered during the first phase, together with the corresponding ids of their  $\leq q$  parents marked as neither  $s_u$  nor  $s_v$ . We can then assume that the algorithm invokes  $Neighbourhood(\cdot)$  on the ids of unmarked level-1 nodes, as querying any other node would not yield any information not already possessed; and we can reinforce it with an oracle that lets it immediately return the correct ranking of the input nodes upon querying  $s_u$  or  $s_v$ . At any point, at most  $q$  level-1 nodes have been marked as neither  $s_u$  nor  $s_v$  by the at most  $q$  local and/or global queries; and thus, since the graph is drawn uniformly at random from  $\mathcal{F}_n$ , at any point the probability of querying  $s_u$  or  $s_v$  is at most  $1/\left(1 + \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil - q\right)$ . Therefore, the overall probability of querying  $s_u$  or  $s_v$  in the second phase is at most  $q/\left(1 + \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil - q\right)$ ,

which for  $q < \frac{1}{2} \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$  is in  $O\left(\frac{q}{n \sqrt{f(n)}}\right)$ , which for any  $f(n) \geq n^{-2/3}$  is in  $O(q \cdot f(n))$ . If neither  $s_u$  nor  $s_v$  are found, the algorithm has once again no information about which input node has the higher PageRank score; therefore, conditioned on the event of reaching the end of the second phase, the probability that the algorithm returns the correct ranking is  $\frac{1}{2}$ . Since the probability of stopping before the end of the second phase and returning the correct ranking is in  $O(q \cdot f(n))$  for any  $q < \frac{1}{2} \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$ , we thus have a total probability bound of  $\frac{1}{2} + O(q \cdot f(n))$  on the event of returning the correct ranking for any  $q < \frac{1}{2} \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$ .

Considering the uniform distribution over the elements of  $\mathcal{F}_n$ , denote by  $\mu = \mu(\mathcal{F}_n)$  the expected number of queries employed by the algorithm, by  $p_q = p_q(\mathcal{F}_n)$  the probability that it employs more than  $q$  queries, and by  $Pr[succcess]$  the probability that it returns the correct ranking of the input nodes.  $Pr[succcess]$  is then upper bounded by  $p_q \cdot 1$  (at best, the algorithm



always returns the correct ranking when using more than  $q$  queries) plus  $(1 - p_q) \leq 1$  times the probability of returning the correct ranking on an element drawn uniformly at random from  $\mathcal{F}_n$  using at most  $q$  queries. Using Markov's inequality to write  $p_q < \frac{\mu}{q}$ , for any  $q < \frac{1}{2} \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$  the following holds:

$$Pr[succcess] \leq \frac{\mu}{q} \cdot 1 + 1 \cdot \left( \frac{1}{2} + O(q \cdot f(n)) \right) = \frac{1}{2} + O\left(\frac{\mu}{q} + q \cdot f(n)\right) \quad (31)$$

For any  $\mu \in o(\frac{1}{f(n)})$  we can thus pick some  $q(n) \in o(\frac{1}{f(n)}) \cap \omega(\mu(\mathcal{F}_n))$  and an  $n_0$  sufficiently large to ensure  $q(n) < \frac{1}{2} \left\lceil \frac{n_0 \sqrt{f(n_0)}}{\alpha} \right\rceil$ , and Equation 31 yields  $Pr[succcess] = \frac{1}{2} + o(1)$ . This holds on average over the elements of  $\mathcal{F}_n$  for any algorithm; and thus, for every given algorithm there exists some  $G \in \mathcal{F}_n$  on which that algorithm must perform an expected  $\Omega(\frac{1}{f(n)})$  queries to return the correct ranking with probability  $\frac{1}{2} + \Omega(1)$ .

For  $\frac{1}{n} \leq f(n) < n^{-\frac{2}{3}}$ , we obtain the generic element of  $\mathcal{F}_n$  by adapting that built for  $f(n) = n^{-\frac{2}{3}}$ . First, remove the self-loops  $(u, u)$  and  $(v, v)$ . Then, let  $k = \left\lceil \log_\alpha \left( f(n_0) \cdot n_0^{2/3} \right) \right\rceil > 0$ , and add  $2k$  nodes  $z_u^1, \dots, z_u^k$  and  $z_v^1, \dots, z_v^k$ , and  $2k$  arcs  $(u, z_u^1), (z_u^1, z_u^2), \dots, (z_u^{k-1}, z_u^k)$  and  $(v, z_v^1), (z_v^1, z_v^2), \dots, (z_v^{k-1}, z_v^k)$ ; and add two self-loops  $(z_u^k, z_u^k), (z_v^k, z_v^k)$ . The size of the graph is clearly still  $\Theta(n_0)$  and can be made arbitrarily large. The maximum outdegree of the graph is  $\Theta(n^{\frac{1}{3}}) = O(\gamma(n))$ . The PageRank scores of  $z_u^k$  and  $z_v^k$  are:

$$P(z_u^k) = \frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{n_0^{2/3}}{\alpha} \right\rceil}{1 + \left\lceil n_0^{1/3} \right\rceil} \right) + \alpha^2 \left\lceil \frac{c\eta n_0^{1/3}}{\alpha^2} \right\rceil \right) \right) \quad (32)$$

$$P(z_v^k) = \frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{n_0^{2/3}}{\alpha} \right\rceil}{1 + \left\lceil n_0^{1/3} \right\rceil} \right) \right) \right) \quad (33)$$

and thus both in  $\Theta\left(\frac{\alpha^k n_0^{1/3}}{n}\right)$ , and since  $\alpha^k = \Theta(f(n_0) \cdot n_0^{2/3})$ , both in  $\Theta\left(\frac{f(n_0) \cdot n_0}{n}\right) = \Theta(f(n))$ .

Their difference  $P(z_u^k) - P(z_v^k)$  is equal to  $\frac{1}{n} \alpha^{k+2} \left\lceil \frac{c\eta n_0^{1/3}}{\alpha^2} \right\rceil$ , and therefore

$$\frac{P(z_u^k) - P(z_v^k)}{P(z_v^k)} = \frac{\frac{1}{n} \alpha^{k+2} \left\lceil \frac{c\eta n_0^{1/3}}{\alpha^2} \right\rceil}{\frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{n_0^{2/3}}{\alpha} \right\rceil}{1 + \left\lceil n_0^{1/3} \right\rceil} \right) \right) \right)} \geq \frac{\alpha^k c\eta n_0^{\frac{1}{3}}}{\left( \frac{1}{1-\alpha} + \alpha^k 2n_0^{\frac{1}{3}} \right)} \quad (34)$$

and by the choice of  $\alpha$  and since  $f(n_0) \geq \frac{1}{n_0}$ , then  $\alpha < \alpha^k n_0^{\frac{1}{3}} \leq 1$  and the last term of Equation 34 is greater than  $\frac{c\alpha\eta}{1/(1-\alpha)+2\alpha}$ , which is greater than or equal to  $\eta$  for any  $c \geq 2 + \frac{1}{\alpha(1-\alpha)}$ .

The family  $\mathcal{F}_n$  consists again of all the graphs isomorphic to  $G$ , and we give in input to the algorithm, in random order, the ids of  $z_u^k$  and  $z_v^k$  in a graph drawn uniformly at random from  $\mathcal{F}_n$ . But returning the correct ranking of  $z_u^k$  and  $z_v^k$  is equivalent to returning that of  $u$  and  $v$ , therefore the  $\Omega(n^{2/3})$  lower bound obtained for  $f(n) = n^{-\frac{2}{3}}$  must hold; and for  $f(n) < n^{-\frac{2}{3}}$ , this bound is equivalent to  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}\right)\right)$ .

We thus have a bound of  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}\right)\right)$  for any  $\gamma(n) > n^{\frac{1}{3}}$  and  $\frac{1}{n} \leq f(n) \leq 1$ .

**Case 1**  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$ . We proceed as follows: first, for any  $f(n) \geq n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$  we exhibit a family  $\mathcal{F}_n$  of arbitrarily large, isomorphic graphs on  $n$  nodes whose generic element has maximum outdegree  $\Theta(\gamma(n))$  and contains two nodes  $u, v$  with PageRank scores in  $\Theta(f(n))$  but not within a factor  $(1 + \eta)$  of each other. We prove that any algorithm, on some element in  $\mathcal{F}_n$ , must perform  $\Omega(\frac{1}{f(n)})$  queries in expectation to decide which among  $u$  and  $v$  has the higher score with probability  $\frac{1}{2} + \Omega(1)$ . We then show how to adapt  $\mathcal{F}_n$  for  $\frac{1}{n} \leq f(n) < n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$  while bringing the above expectation to  $\Omega(n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}})$ , thus obtaining a lower bound of  $\Omega(\min(\frac{1}{f(n)}, n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}))$  for any  $\frac{1}{n} \leq f(n) \leq 1$ .

To build a generic element  $G$  of  $\mathcal{F}_n$  for  $f(n) \geq n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$ , consider an arbitrary positive integer  $n_0$  (a rough approximation of the final number of nodes in  $G$ ). The nodes of  $G$  (see Figure 3) can be divided into three levels, plus some additional isolated nodes (see below). Level 0 consists of 2 nodes  $u, v$  and  $2 \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil \lceil \gamma(n_0) \rceil$  nodes  $w_0^0, w_0^1, \dots$ ; each node in this level has (only) a self loop. Level 1 consists of 2 nodes  $s_u, s_v$  and  $2 \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil$  nodes  $w_1^0, w_1^1, \dots$ . The sole outgoing arcs of  $s_u$  and  $s_v$  are, respectively,  $(s_u, u)$  and  $(s_v, v)$ ; while  $(w_1^j, u)$  is an arc for  $0 \leq j < \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil$ , and  $(w_1^j, v)$  is an arc for  $\left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil \leq j < 2 \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil$ . Furthermore, each node  $w_1^j$  has outgoing arcs  $(w_1^j, w_0^{\lceil \gamma(n_0) \rceil j}), \dots, (w_1^j, w_0^{\lceil \gamma(n_0) \rceil (j+1)-1})$ . Finally, level 2 consists of  $\left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil$  nodes  $w_2^0, w_2^1, \dots$  having outgoing arcs  $(w_2^0, s_u), (w_2^1, s_u), \dots$ , towards  $s_u$ , for a constant  $c > 0$  to be determined later. The family  $\mathcal{F}_n$  consists of all the graphs isomorphic to  $G$ . In particular, in an element drawn uniformly at random from  $\mathcal{F}_n$ , each node is identified by a random integer in  $\{1, \dots, n\}$ ; but we will keep denoting by  $u$  and  $v$  the (only) two level-0 nodes each having  $1 + \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil \geq 2$  incoming arcs from level-1 nodes, and by  $s_u$  and  $s_v$  the (only) two level-1 nodes whose sole arc points towards  $u$  and  $v$  respectively, so that  $s_u$  is the node pointed by all the level-2 nodes.

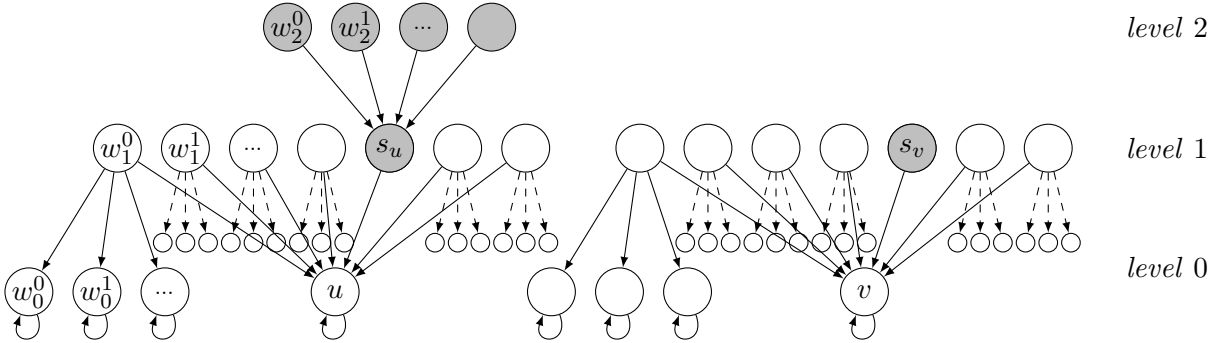


Figure 3: A generic graph  $G$  of the family  $\mathcal{F}_n$  for  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$  and  $f(n) \geq n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$ . All level-1 nodes except  $s_u$  and  $s_v$  have  $\lceil \gamma(n_0) \rceil$  additional children besides  $u$  or  $v$  but, for simplicity, we have drawn these children in normal size only for the leftmost node. Again, in order to distinguish the two subgraphs, and thus  $u$  from  $v$ , one must find at least one of the nodes depicted in grey.

The number of nodes in  $G$  is  $n = 4 + 2 \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil \lceil \gamma(n_0) \rceil + 2 \left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil + \left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil$ , which is always between  $4\sqrt{n_0}$  and  $\frac{16+2c\eta}{\alpha^2} n_0$ . We then complete  $G$  by adding isolated nodes until  $n = \frac{16+2c\eta}{\alpha^2} n_0 \in \Theta(n_0)$ , which one can make arbitrarily large by increasing  $n_0$ . The maximum

outdegree of the graph is  $1 + \lceil \gamma(n_0) \rceil = \Theta(\gamma(n))$ . The PageRank scores of  $u$  and  $v$  are:

$$P(u) = \frac{1}{n} \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) + \alpha^2 \left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil \right) \quad (35)$$

$$P(v) = \frac{1}{n} \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) \right) \quad (36)$$

and both are in  $\Theta\left(\frac{1}{n} \left( \sqrt{\frac{n_0}{\gamma(n_0)}} + n_0 f(n_0) \right)\right)$ , and since  $n = \Theta(n_0)$  and  $f(n) \geq n^{-\frac{1}{2}} \gamma(n)^{-\frac{1}{2}}$ , both in  $\Theta(f(n))$ . Their difference  $P(u) - P(v)$  is equal to  $\frac{1}{n} \alpha^2 \left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil$ , and therefore:

$$\frac{P(u) - P(v)}{P(v)} = \frac{\frac{1}{n} \alpha^2 \left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil}{\frac{1}{n} \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) \right)} \geq \frac{c\eta n_0 f(n_0)}{1 + \alpha + \frac{2\sqrt{n_0 \gamma(n_0)}}{\gamma(n_0)}} = \frac{c\eta}{\frac{1+\alpha}{n_0 f(n_0)} + \frac{2}{f(n_0) \sqrt{n_0 \gamma(n_0)}}} \quad (37)$$

and since  $f(n) \geq n^{-\frac{1}{2}} \gamma(n)^{-\frac{1}{2}} \geq \frac{1}{n}$ , for  $c \geq 4$  the last term is greater than  $\eta$  for any  $\eta > 0$ , any  $\alpha \in (0, 1)$ , and any  $n_0 > 0$ .

Consider now a generic algorithm that receives in input, in random order, the ids of nodes  $u$  and  $v$  in a graph drawn uniformly at random from  $\mathcal{F}_n$ , and must decide which one has the higher PageRank score using only natural exploration queries. We apply the same argument deployed for the case  $\gamma(n) > n^{\frac{1}{3}}$  (see above), with now  $\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil = \Theta(n^{\frac{1}{2}} \gamma(n)^{\frac{1}{2}})$  level-1 nodes and  $\left\lceil \frac{c\eta}{\alpha^2} n_0 f(n_0) \right\rceil = \Theta(n f(n))$  level-2 nodes; note that the presence of isolated nodes only reduces the probability that a global query reveals useful information, and do not influence the probability that a local query reveals useful information. This gives a lower bound of  $\Omega(\min(n^{\frac{1}{2}} \gamma(n)^{\frac{1}{2}}, \frac{1}{f(n)}))$  queries to bring the probability of success on  $\mathcal{F}_n$  to  $\frac{1}{2} + \Omega(1)$ .

For  $\frac{1}{n} \leq f(n) < n^{-\frac{1}{2}} \gamma(n)^{-\frac{1}{2}}$ , we obtain the generic element of  $\mathcal{F}_n$  by adapting the graph built for  $f(n) = n^{-\frac{1}{2}} \gamma(n)^{-\frac{1}{2}}$ . First, remove the self-loops  $(u, u)$  and  $(v, v)$ . Then let  $k = \left\lceil \log_\alpha(f(n_0) \cdot n_0^{1/2} \gamma(n_0)^{1/2}) \right\rceil > 0$ , and add  $2k$  nodes  $z_u^1, \dots, z_u^k$  and  $z_v^1, \dots, z_v^k$ , and  $2k$  arcs  $(u, z_u^1), (z_u^1, z_u^2), \dots, (z_u^{k-1}, z_u^k)$  and  $(v, z_v^1), (z_v^1, z_v^2), \dots, (z_v^{k-1}, z_v^k)$ ; and add two self-loops  $(z_u^k, z_u^k)$ ,  $(z_v^k, z_v^k)$ . The size of the graph is clearly still  $\Theta(n_0)$  and can be made arbitrarily large. The maximum outdegree of the graph is still  $\Theta(\gamma(n))$ . The PageRank scores of  $z_u^k$  and  $z_v^k$  are:

$$P(z_u^k) = \frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) + \alpha^2 \left\lceil \frac{c\eta}{\alpha^2} \sqrt{\frac{n_0}{\gamma(n_0)}} \right\rceil \right) \right) \quad (38)$$

$$P(z_v^k) = \frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0 \gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) \right) \right) \quad (39)$$

and thus both in  $\Theta\left(\frac{\alpha^k n_0^{1/2} \gamma(n_0)^{-1/2}}{n}\right)$ , and since  $\alpha^k = \Theta(f(n_0) \cdot n_0^{1/2} \gamma(n_0)^{1/2})$ , both in  $\Theta\left(\frac{f(n_0) \cdot n_0}{n}\right) =$

$\Theta(f(n))$ . Their difference  $P(z_u^k) - P(z_v^k)$  is equal to  $\frac{1}{n}\alpha^{k+2} \left\lceil \frac{c\eta}{\alpha^2} \sqrt{\frac{n_0}{\gamma(n_0)}} \right\rceil$ , and therefore

$$\frac{P(z_u^k) - P(z_v^k)}{P(z_v^k)} = \frac{\frac{1}{n}\alpha^{k+2} \left\lceil \frac{c\eta}{\alpha^2} \sqrt{\frac{n_0}{\gamma(n_0)}} \right\rceil}{\frac{1}{n} \left( \sum_{i=0}^{k-1} \alpha^i + \alpha^k \left( 1 + \alpha \left( 1 + \frac{\left\lceil \frac{\sqrt{n_0\gamma(n_0)}}{\alpha} \right\rceil}{1 + \lceil \gamma(n_0) \rceil} \right) \right) \right)} \geq \frac{\alpha^k c\eta \sqrt{\frac{n_0}{\gamma(n_0)}}}{\left( \frac{1}{1-\alpha} + \alpha^k 2\sqrt{\frac{n_0}{\gamma(n_0)}} \right)} \quad (40)$$

and by the choice of  $k$  and since  $f(n_0) \geq \frac{1}{n_0}$ , then  $\alpha < \alpha^k \sqrt{\frac{n_0}{\gamma(n_0)}} \leq 1$  and the last term of Equation 40 is greater than  $\frac{c\alpha\eta}{1/(1-\alpha)+2\alpha}$ , which is greater than or equal to  $\eta$  for any  $c \geq 2 + \frac{1}{\alpha(1-\alpha)}$ .

The family  $\mathcal{F}_n$  consists again of all the graphs isomorphic to  $G$ , and we give in input to the algorithm, in random order, the ids of  $z_u^k$  and  $z_v^k$  in a graph drawn uniformly at random from  $\mathcal{F}_n$ . But returning the correct ranking of  $z_u^k$  and  $z_v^k$  is equivalent to returning that of  $u$  and  $v$ , therefore the  $\Omega(n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}})$  lower bound obtained for  $f(n) = n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$  must hold; and for  $f(n) < n^{-\frac{1}{2}}\gamma(n)^{-\frac{1}{2}}$ , this bound is equivalent to  $\Omega(\min(n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}, \frac{1}{f(n)}))$ .

We thus have a bound of  $\Omega(\min(n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}, \frac{1}{f(n)}))$  for any  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$  and  $\frac{1}{n} \leq f(n) \leq 1$ .

**Generalization of  $f(n)$  and  $\gamma(n)$ .** To remove the assumption that  $f(n)$  and  $\gamma(n)$  are respectively in  $\Theta(f(n_0))$  and  $\Theta(\gamma(n_0))$  whenever  $n = \Theta(n_0)$ , one must build the graphs as a function of  $n$  instead of  $n_0$ . Crucially, we must then properly rescale the number of nodes at each level in order to ensure that the graph has size exactly  $n$  while still guaranteeing that the scores of  $u$  and  $v$  are not within a factor  $(1 + \eta)$  of each other. We show how to perform this “rescale” operation only for the case  $\gamma(n) > n^{\frac{1}{3}}$  (see above); the case  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$  is completely analogous. For  $n^{-\frac{2}{3}} \leq f(n) \leq 1$ , consider three arbitrarily small, positive reals  $\lambda_0, \lambda_1, \lambda_2$ . The graph has the same structure as shown above (see Figure 2), but each subgraph has now  $\lceil \lambda_1 n \sqrt{f(n)} \rceil$  level-1 nodes (not counting  $s_u$  and  $s_v$ ) each having  $\left\lceil \frac{\lambda_0}{\sqrt{f(n)}} \right\rceil$  additional children besides  $u$  or  $v$  (except for  $s_u$  and  $s_v$ ), and  $\lceil \lambda_2 n f(n) \rceil$  level-2 nodes. The size of the graph is  $\Theta(\lambda_0 \lambda_1 n + \lambda_2 n f(n)) = O(n(\lambda_0 \lambda_1 + \lambda_2))$ ; we can thus make it exactly equal to  $n$  by choosing sufficiently small lambdas and then adding isolated nodes if necessary. Note that now both the scores of the nodes and the lower bound still satisfy the thesis, even if  $f(\Theta(n)) \notin \Theta(n)$ . We can further choose  $\lambda_0$  and  $\lambda_1$  such that also  $\frac{\lambda_1}{\lambda_0}$  is sufficiently small; and this ensures that the score contribution of the level-2 nodes is large enough to have the scores not within a factor  $(1 + \eta)$  of each other, for any  $n$  larger than a constant factor depending only on  $\eta, \alpha, \lambda_0, \lambda_1, \lambda_2$ . Finally, we must ensure that the contribution of the level-2 nodes is sufficient to keep the scores differing by more than  $(1 + \eta)$  when adapting the graph for  $f(n) < n^{-\frac{2}{3}}$ , even when  $f(n) = \Theta(\frac{1}{n})$  (i.e. when the contribution of the nodes nearest to  $z_u^k$  and  $z_v^k$  is also in  $\Theta(\frac{1}{n})$  and may bring  $P(z_u^k)$  and  $P(z_v^k)$  too close). To do this, we simply reduce the length of the chain of nodes leading to  $z_u^k$  and  $z_v^k$  by a sufficiently large constant additive factor (note that this does not change the value of the scores asymptotically), until the score contribution from the level-2 nodes becomes sufficiently large.

**Assembling the lower bound.** For any  $\gamma(n) > n^{\frac{1}{3}}$  and any  $\frac{1}{n} \leq f(n) \leq 1$ , we have proven a lower bound of  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}\right)\right) = \Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}, n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}\right)\right)$  queries; and for any  $1 \leq \gamma(n) \leq n^{\frac{1}{3}}$  and any  $\frac{1}{n} \leq f(n) < 1$ , we have also proven a lower bound of  $\Omega(\min(n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}, \frac{1}{f(n)})) = \Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}, n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}\right)\right)$  queries. Therefore, the general lower bound remains  $\Omega\left(\min\left(\frac{1}{f(n)}, n^{2/3}, n^{\frac{1}{2}}\gamma(n)^{\frac{1}{2}}\right)\right)$ , which concludes the proof.

### A.3 Proof of Lemma 1

The routine *SampleNode()* is formally defined as follows:

---

**Algorithm 1** *SampleNode()*


---

```

current_node  $\leftarrow$  RandomNode()
loop
  with probability  $(1 - \alpha)$  return current_node
  current_node  $\leftarrow$  RandomChild(current_node)
  if current_node =  $\emptyset$  then current_node  $\leftarrow$  RandomNode()
end loop

```

---

We first prove that *SampleNode()* returns node  $v$  with probability  $P(v)$ . Consider the random walk used in Subsection 1.1 to define PageRank in terms of an abstract surfer. We can imagine the surfer selects at any node, whether dangling or not, with probability  $(1 - \alpha)$  a *random jump* leading to a node chosen uniformly at random, and with probability  $\alpha$  either a *virtual arc* leading to a node chosen uniformly at random if the current node is dangling, or an actual arc leading to a child (chosen uniformly at random) of the current node otherwise. The stationary probability of the latest jump being  $\tau$  steps in the past (i.e. of having followed exactly  $\tau$  arcs, whether virtual or actual, after it) is  $(1 - \alpha)\alpha^\tau$ . Since *SampleNode()* follows, starting from a node chosen uniformly at random, exactly  $\tau$  arcs (virtual or actual) with probability  $(1 - \alpha)\alpha^\tau$ , it returns node  $v$  with probability equal to  $P(v)$ .

We now prove the bounds on the number of queries employed by *SampleNode()*. Each time *SampleNode()* goes through the first instruction of the loop, it terminates immediately with an independent probability  $(1 - \alpha)$ ; and, by the  $j$ -th time, it has issued at most  $1 + 2(j - 1) \leq 2j$  queries. Thus, *SampleNode()* terminates after issuing at most  $\frac{2}{1 - \alpha}$  queries in expectation. If *SampleNode()* issues more than  $\frac{2(1 + \Delta)m}{1 - \alpha}$  queries, then it goes through the first instruction in the loop more than  $\frac{(1 + \Delta)m}{1 - \alpha}$  times; the probability of this happening over  $m$  calls is the probability that that instruction makes *SampleNode()* terminate less than  $m = (1 - \frac{\Delta}{1 + \Delta}) \cdot (1 + \Delta)m$  out of  $\frac{(1 + \Delta)m}{1 - \alpha}$  times. By a simple probability bound (see Appendix A.1), this probability is less than 
$$e^{-\frac{(1 + \Delta)m(\frac{\Delta}{1 + \Delta})^2}{2}} = e^{-\frac{m}{2} \cdot \frac{\Delta^2}{1 + \Delta}}.$$

### A.4 Proof of Lemma 2

Consider the random walk used in Subsection 1.1 to define PageRank in terms of an abstract surfer. We can imagine the surfer selects at any node, whether dangling or not, with probability  $(1 - \alpha)$  a *random jump* leading to a node chosen uniformly at random, and with probability  $\alpha$  either a *virtual arc* leading to a node chosen uniformly at random if the current node is dangling, or an actual arc leading to a child (chosen uniformly at random) of the current node otherwise.

Denote by  $E_{v,t}$  the event that the surfer is at  $v$  at time  $t$ , and let  $\bar{t} \leq t$  be the latest time before or coinciding with  $t$  when the surfer arrives at a node with a random jump.  $E_{v,t}$  takes place if either one of two disjoint events  $E_{v,t}^{\in \bar{G}}$  and  $E_{v,t}^{\notin \bar{G}}$  takes place.  $E_{v,t}^{\in \bar{G}}$  is the event that the random surfer reaches  $v$  at  $t$  without ever being in a node outside  $\bar{G}$  between  $\bar{t}$  and  $t$ .  $E_{v,t}^{\notin \bar{G}}$  is the event that the random surfer reaches  $v$  at  $t$  being in at least one node outside  $\bar{G}$  between  $\bar{t}$  and  $t$ . Note that the last such node must either have an outgoing arc in  $F(\bar{G})$ , or be a dangling node in  $G \setminus \{v\}$  (as, by hypothesis, these are the dangling nodes outside  $\bar{G}$ ) – and that after leaving it, the random surfer must follow a path entirely in  $\bar{G}$ .

If  $\text{outdegree}(v) > 0$ , then  $E_{v,t}^{\in \bar{G}}$  takes place if and only if the random surfer follows only arcs of  $\bar{G}$  between  $\bar{t}$  and  $t$ . Recall that the probability of the latest random jump no later than  $t$  being at  $\bar{t}$  is  $(1 - \alpha)\alpha^{t - \bar{t}}$ , and that every node has the same probability  $\frac{1}{n}$  of being the destination

of such a jump. We can then write:

$$\begin{aligned} P(E_{v,t}^{\in \bar{G}}) &= \sum_{\bar{t} \leq t} \left( (1-\alpha) \alpha^{t-\bar{t}} \sum_{|\pi_{w,v}^{\bar{G}}|=t-\bar{t}} \left( \frac{1}{n} \prod_{(u,u') \in \pi_{w,v}^{\bar{G}}} \frac{1}{\text{outdegree}(u)} \right) \right) \\ &= \sum_{w \in \bar{G}} \left( \frac{1-\alpha}{n} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) \end{aligned} \quad (41)$$

If instead  $\text{outdegree}(v) = 0$ , then  $E_{v,t}^{\in \bar{G}}$  also includes the event that the random surfer follows one or more of  $v$ 's "virtual arcs" between  $\bar{t}$  and  $t$ . Denote by  $\hat{t} < t$  the latest time before  $t$  when the surfer is in  $v$ ; then at  $\hat{t} + 1$  the surfer has probability  $\alpha/n$  of arriving at any given node  $w \in \bar{G}$  via one of  $v$ 's virtual arcs, and probability  $\mathfrak{U}_{w,v}^{\bar{G}}$  of next following a path  $\pi_{w,v}^{\bar{G}}$  of length  $t - \hat{t} - 1$  from  $w$  to  $v$  in  $\bar{G}$ . Thus:

$$P(E_{v,t}^{\in \bar{G}}) = \sum_{w \in \bar{G}} \left( \frac{1-\alpha}{n} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) + \sum_{\hat{t} < t} P(E_{v,\hat{t}}^{\in \bar{G}}) \left( \sum_{|\pi_{w,v}^{\bar{G}}|=t-\hat{t}-1} \frac{\alpha}{n} \mathfrak{U}_{w,v}^{\bar{G}} \right) \quad (42)$$

But since the stationary probability  $P(E_{v,\hat{t}}^{\in \bar{G}})$  is the same for  $\hat{t} = t$ , we obtain:

$$\begin{aligned} P(E_{v,t}^{\in \bar{G}}) &= \sum_{w \in \bar{G}} \left( \frac{1-\alpha}{n} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) + P(E_{v,t}^{\in \bar{G}}) \frac{\alpha}{n} \sum_{\hat{t} < t} \left( \sum_{|\pi_{w,v}^{\bar{G}}|=t-\hat{t}-1} \mathfrak{U}_{w,v}^{\bar{G}} \right) \\ &= \sum_{w \in \bar{G}} \left( \frac{1-\alpha}{n} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) + P(E_{v,t}^{\in \bar{G}}) \frac{\alpha}{n} \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \\ &= \sum_{w \in \bar{G}} \left( \frac{1-\alpha}{n} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) \cdot \frac{1}{1 - \frac{\alpha}{n} \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}}} \end{aligned} \quad (43)$$

We can analyse the event  $E_{v,t}^{\notin \bar{G}}$  in a similar way. If  $\text{outdegree}(v) > 0$ , then  $E_{v,t}^{\notin \bar{G}}$  takes place if and only if the random surfer follows only arcs of  $\bar{G}$  after being one last time, at some  $t' < t$ , in a node outside  $\bar{G}$  (i.e. either a node with an outgoing arc in  $F(\bar{G})$  or a dangling node in  $G \setminus \{v\}$ ). We can then write:

$$\begin{aligned} P(E_{v,t}^{\notin \bar{G}}) &= \sum_{t' < t} \sum_{(u,w) \in F(\bar{G})} \left( P(E_{u,t'}) \frac{\alpha}{\text{outdegree}(u)} \sum_{|\pi_{w,v}^{\bar{G}}|=t-t'-1} \mathfrak{U}_{w,v}^{\bar{G}} \right) \\ &\quad + \sum_{t' < t} \sum_{\substack{u \in G \setminus \{v\}: \\ \text{outdegree}(u)=0}} P(E_{u,t'}) \frac{\alpha}{n} \left( \sum_{|\pi_{w,v}^{\bar{G}}|=t-t'-1} \mathfrak{U}_{w,v}^{\bar{G}} \right) \\ &= \sum_{(u,w) \in F(\bar{G})} \left( \alpha \frac{P(u)}{\text{outdegree}(u)} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) + \sum_{\substack{u \in G \setminus \{v\}: \\ \text{outdegree}(u)=0}} \left( \alpha \frac{P(u)}{n} \cdot \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \right) \end{aligned} \quad (44)$$

Conversely, if  $\text{outdegree}(v) = 0$ , then  $E_{v,t}^{\notin \bar{G}}$  also includes the event of the random surfer following one or more times a virtual arc from  $v$  to  $\bar{G}$  after entering  $\bar{G}$  for the last time, and then following

only arcs of  $\bar{G}$  until time  $t$ . The same analysis leading to Equation 43 then yields:

$$P(E_{v,t}^{\bar{G}}) = \left( \sum_{(u,w) \in F(\bar{G})} \left( \alpha \frac{P(u)}{\text{outdegree}(u)} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) + \sum_{\substack{u \in G \setminus \{v\}: \\ \text{outdegree}(u)=0}} \left( \alpha \frac{P(u)}{n} \cdot \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \right) \right) \cdot \frac{1}{1 - \frac{\alpha}{n} \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}}} \quad (45)$$

Combining Equations 41, 43, 44, 45, and setting

$$\mu_v^{\bar{G}} = \begin{cases} 1 & \text{if } \text{outdegree}(v) > 0 \\ 1 / \left( 1 - \frac{\alpha}{n} \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \right) & \text{if } \text{outdegree}(v) = 0 \end{cases} \quad (46)$$

we finally obtain:

$$P^{\bar{G}}(v) = \mu_v^{\bar{G}} \cdot \left( \sum_{w \in \bar{G}} \mathfrak{U}_{w,v}^{\bar{G}} \left( \frac{1-\alpha}{n} + \sum_{\substack{u \in G \setminus \{v\}: \\ \text{outdegree}(u)=0}} \alpha \frac{P(u)}{n} \right) + \sum_{(u,w) \in F(\bar{G})} \left( \alpha \frac{P(u)}{\text{outdegree}(u)} \cdot \mathfrak{U}_{w,v}^{\bar{G}} \right) \right) \quad (47)$$

## A.5 Formal construction of the sequence of graphs $G_1, G_2, \dots$

Let us first consider the case  $m = 1$ . Choose as  $u_1$  the parent  $u$  of  $v$  with the largest  $1/\gamma_u$ , breaking ties arbitrarily. Equation 15 is trivially satisfied (since there is only one positive index  $i \leq m$ ) regardless of the value assigned to  $\beta_1^1$ . It is immediate to verify that setting  $\beta_1^1 = 0$  satisfies Equation 16, but violates Equation 17 unless the largest multiplier  $\frac{1}{\gamma_u}$  is shared by multiple parents of  $v$  (or unless  $u_1$  is  $v$ 's only ancestor). However, since  $\beta_0^1 = (1 - \beta_1^1)$ , setting  $\beta_1^1 = 1$  would yield with  $m = 1$  for all  $i'$ :

$$\frac{1}{\gamma_{u_1}} \cdot \sum_{j=j_1}^{1-1} \beta_j^m \cdot c_1^{G_j} = 0 \leq \frac{1}{\gamma_{u_{i'}}} \cdot \sum_{j=j_{i'}}^m \beta_j^m \cdot c_{i'}^{G_j} \quad (48)$$

and for the continuity of all products  $\frac{1}{\gamma_{u_i}} \cdot \beta_j^1 \cdot c_i^{G_j}$  as a function of  $\beta_1^1$ , there exists a  $\beta_1^1 \in [0, 1]$  that satisfies both Equation 16 and Equation 17.

Assume now that Equations 15, 16 and 17 are satisfied immediately after adding node  $u_h$  with an appropriate choice of  $\beta_h^h$ . Let  $u_{h+1}$  be a node satisfying Equation 17 for  $m = h$  (i.e.  $\forall i \leq h$ , we have  $\frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} \beta_j^h \cdot c_i^{G_j} = \frac{1}{\gamma_{u_{h+1}}} \cdot \sum_{j=j_{h+1}}^h \beta_j^h \cdot c_{h+1}^{G_j}$ ). This automatically satisfies Equation 15 for any choice of  $\beta_{h+1}^{h+1}$ , and also satisfies Equation 16 for all “sufficiently small” values of  $\beta_{h+1}^{h+1}$  and in particular for  $\beta_{h+1}^{h+1} = 0$  (in which case the weighted coefficients of all  $x_u$  coincide with those for  $m = h$ ). We can then make an argument identical to that made for  $m = 1$ , noting that for  $\beta_{h+1}^{h+1} = 1$  and  $1 \leq i \leq h+1$  we have:

$$\frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} \beta_j^{h+1} \cdot c_i^{G_j} = \frac{1}{\gamma_{u_i}} \cdot \sum_{j=j_i}^{i-1} (1 - \beta_{h+1}^{h+1}) \beta_j^h \cdot c_i^{G_j} = 0 \leq \frac{1}{\gamma_{u_{i'}}} \cdot \sum_{j=j_{i'}}^{h+1} \beta_j^{h+1} \cdot c_{i'}^{G_j} \quad (49)$$

to conclude from the continuity of all products  $\frac{1}{\gamma_{u_i}} \cdot \beta_j^{h+1} \cdot c_i^{G_j}$  as a function of  $\beta_{h+1}^{h+1}$  that there exists a  $\beta_{h+1}^{h+1} \in [0, 1]$  satisfying both Equation 16 and Equation 17 for  $m = h+1$ .

## A.6 A note on Equation 22

If we knew the outdegree of every  $u_i$ , setting  $\gamma_{u_i} = \text{outdegree}(u_i)$ , for  $\bar{n} = m = \Theta(n^{\frac{1}{2}})$  we would then have  $E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \frac{\bar{n} \cdot m}{n} = \Theta(1)$  (and the constant involved could be made arbitrarily large taking sufficiently large  $\bar{n} = m$ ). More in general, if we knew for every  $u$  that  $\text{outdegree}(u)$  fell within some interval  $[\text{mindeg}(u), \text{maxdeg}(u)]$ , we could set each  $\gamma_{u_i}$  to e.g.  $\text{mindeg}(u_i)$  to obtain  $E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \frac{\bar{n} \cdot m}{n} \min_u \frac{\text{mindeg}(u)}{\text{maxdeg}(u)}$  – which could be made  $\Theta(1)$ , with an arbitrarily large hidden constant, by choosing sufficiently large  $\bar{n} = m = \Theta\left(n^{\frac{1}{2}} \sqrt{\max_u \frac{\text{maxdeg}(u)}{\text{mindeg}(u)}}\right)$  (the same result could be obtained setting instead every  $\gamma_{u_i}$  to  $\text{maxdeg}(u_i)$ ). Note that in many networks, even though the maximum indegree of a node could be  $\Theta(n)$ , the maximum *outdegree* of a node is limited. For example, in an online social network a single user can be consistently followed by millions of people, but will hardly be able to consistently follow more than a few hundred other people. Similarly, virtually all scientific articles cite less than 100 other articles. If we can assume such an outdegree bound exists, we may correspondingly reduce the total number of queries required by our algorithm.

## A.7 A bound for $E[r_{G_m}^{\bar{n}}]$

To obtain a bound on the distribution of  $E[r_{G_m}^{\bar{n}}]$ , we analyse separately the numerator and denominator of the rightmost term in the product forming the right-hand side of Equation 22. A crucial observation in both cases is that  $\psi_{u_i}^h$  and  $\psi_{u_{i'}}^{h'}$  may be positively or negatively correlated if  $h = h'$  (e.g. if  $u_i$  and  $u_{i'}$  share all children or none, respectively), but are independent for  $h \neq h'$ .

Let us begin with the numerator,  $\sum_{i=1}^m \frac{\gamma_{u_i}}{\text{outdegree}(u_i)}$ , recalling that by construction  $\text{outdegree}(u_i) > 0$  for every  $i = 1, \dots, m$ . If  $\text{outdegree}(u) \leq \frac{cn \log(n)}{\bar{n}}$ , then  $\frac{g(u)}{\text{outdegree}(u)} > 1$  with probability 1. If  $\text{outdegree}(u) > \frac{cn \log(n)}{\bar{n}}$ , the sum  $\sum_{h=1}^{\bar{n}} \psi_u^h$  of  $\bar{n}$  independent random variables with support in  $[0, 1]$  has expectation  $\frac{\bar{n}}{n} \text{outdegree}(u) > c \log(n)$ . Thus (see Appendix A.1) the probability that it fails to reach half its expectation is less than  $e^{-\frac{c \log(n)}{8}} = n^{-\frac{c}{8 \log_e(2)}}$ ; this is also an upper bound on the probability that  $\frac{n}{\bar{n}} \sum_{h=1}^{\bar{n}} \psi_u^h$  fails to reach half its expectation  $\text{outdegree}(u)$ , and thus an upper bound on the probability that  $g(u)$ , which is always strictly larger than  $\frac{n}{\bar{n}} \sum_{h=1}^{\bar{n}} \psi_u^h$ , fails to reach  $\frac{\text{outdegree}(u)}{2}$ . Taking a union bound on all  $u$  in  $G$ , for all sufficiently large  $c$  (such that  $n^{-\frac{c}{8 \log_e(2)}} \cdot n = n^{-\Theta(c)}$ ) we have that  $\sum_{i=1}^m \frac{\gamma_{u_i}}{\text{outdegree}(u_i)} < \sum_{i=1}^m \frac{1}{2} = \frac{m}{2}$  with probability at most  $n^{-\Theta(c)}$ .

Let us turn to the denominator,  $\frac{\gamma_{u_\ell}}{\text{outdegree}(u_\ell)} \leq \max_u \frac{g(u)}{\text{outdegree}(u)}$ . The probability that  $\sum_{h=1}^{\bar{n}} \psi_u^h$  exceeds  $\frac{cn \log(n)}{\bar{n}} E\left[\sum_{h=1}^{\bar{n}} \psi_u^h\right] = c \log(n) \cdot \text{outdegree}(u)$  is then (see Appendix A.1, remembering that for  $\Delta \geq 1$  we have that  $\frac{\Delta^2 \mu}{2 + \Delta} \geq \frac{\Delta + 1}{3} \frac{\Delta \mu}{\Delta} = \frac{(1 + \Delta)\mu}{6}$ ) at most  $e^{-\frac{c \log(n) \cdot \text{outdegree}(u)}{6}} \leq n^{-\frac{c}{6 \log_e(2)}}$ . This is then also an upper bound to the probability that  $\frac{n}{\bar{n}} \sum_{h=1}^{\bar{n}} \psi_u^h$  exceeds  $\frac{n}{\bar{n}} c \log(n) \cdot \text{outdegree}(u)$  and thus to the probability that  $g(u)$  exceeds  $\frac{cn \log(n)}{\bar{n}} + \frac{n}{\bar{n}} c \log(n) \cdot \text{outdegree}(u) \leq 2 \frac{cn \log(n)}{\bar{n}} \text{outdegree}(u)$ . Taking again a union bound on all  $u$  in  $G$ , for all sufficiently large  $c$  (such that  $n^{-\frac{c}{6 \log_e(2)}} \cdot n = n^{-\Theta(c)}$ ) the probability that  $\max_u \left(\frac{g(u)}{\text{outdegree}(u)}\right)$  exceeds  $2 \frac{cn \log(n)}{\bar{n}}$  is  $n^{-\Theta(c)}$ .

Taking one more union bound (on the probability of the numerator being “too small” and of the denominator being “too large”) with probability  $1 - n^{-\Theta(c)}$  Equation 19 becomes:

$$E[r_{G_m}^{\bar{n}}] \geq (1 - \alpha) \frac{\bar{n}}{n} \cdot \frac{m/2}{2cn \log(n)/\bar{n}} = (1 - \alpha) \frac{m \bar{n}^2}{4cn^2 \log(n)} \quad (50)$$



## References

- [1] A. Anagnostopoulos, R. Kumar, M. Mahdian, E. Upfal, and F. Vandin. Algorithms on evolving graphs. In *Proc. of ITCS*, pages 149–160. 2012.
- [2] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, V. Mirrokni, and S.-H. Teng. Local computation of PageRank contributions. *Internet Mathematics*, 5(1–2):23–45, 2008.
- [3] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *Proc. of IEEE FOCS*, pages 475–486. 2006.
- [4] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. In *Proc. of ACM STOC*, pages 30–41. 1977.
- [5] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte Carlo methods in PageRank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis*, 45(2):890–904, 2007.
- [6] B. Bahmani, R. Kumar, M. Mahdian, and E. Upfal. PageRank on an evolving graph. In *Proc. of ACM KDD*, pages 24–32. 2012.
- [7] Z. Bar-Yossef and L.-T. Mashiach. Local approximation of PageRank and reverse PageRank. In *Proc. of ACM CIKM*, pages 279–288. 2008.
- [8] Z. Bar-Yossef and L.-T. Mashiach. Local approximation of PageRank and reverse PageRank. In *Proc. of ACM SIGIR*, pages 865–866. 2008.
- [9] P. Berkhin. A survey on PageRank computing. *Internet Mathematics*, 2(1):73–120, 2005.
- [10] C. Borgs, M. Brautbar, J. T. Chayes, and S.-H. Teng. Multi-scale matrix sampling and sublinear-time PageRank computation. *CoRR*, abs/1202.2771, 2013.
- [11] C. Borgs, M. Brautbar, J. T. Chayes, and S.-H. Teng. A sublinear time algorithm for PageRank computations. In *Proc. of WAW*, pages 41–53. 2012.
- [12] M. Brautbar and M. Kearns. Local algorithms for finding interesting individuals in large networks. In *Proc. of ICS*, pages 188–199. 2010.
- [13] M. Bressan, E. Peserico, and L. Pretto. The power of local information in PageRank. Technical report, INRIA/Université Paris-Sud. <http://www.lri.fr/~bressanm/localpr.pdf>. 2013.
- [14] M. Bressan and L. Pretto. Local computation of PageRank: the ranking side. In *Proc. of ACM CIKM*, pages 631–640. 2011.
- [15] S. Brin and L. Page. The anatomy of a large scale hypertextual Web search engine. In *Proc. of WWW*. 1998.
- [16] Y.-Y. Chen, Q. Gan, and T. Suel. Local methods for estimating PageRank values. In *Proc. of ACM CIKM*, pages 381–389. 2004.
- [17] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005.
- [18] F. Geerts, H. Mannila, and E. Terzi. Relational link-based ranking. In *Proc. of VLDB*, pages 552–563. 2004.

- [19] D. Gleich and M. Polito. Approximating personalized PageRank with minimal use of web graph data. *Internet Mathematics*, 3(3):257–294, 2007.
- [20] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. of VLDB*, pages 576–587. 2004.
- [21] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of WWW*, pages 508–516. 2003.
- [22] C. E. Lee, A. Ozdaglar, and D. Shah. Computing the stationary distribution, locally. In *Proc. of NIPS*, pages 1376–1384. 2013.
- [23] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert. GeneRank: Using search engine technology for the analysis of microarray experiments. *BMC Bioinformatics*, 6:233, 2005.
- [24] D. Saez-Trumper, G. Comarela, V. Almeida, R. Baeza-Yates, and F. Benevenuto. Finding trendsetters in information networks. In *Proc. of ACM KDD*, pages 1014–1022. 2012.
- [25] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.
- [26] P. Tarau, R. Mihalcea, and E. Figa. Semantic document engineering with WordNet and PageRank. In *Proc. of ACM SAC*, pages 782–786. 2005.
- [27] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2007.